الجمهورية الجزائرية الديمقراطية الشعبية
Republique Algerienne Democratique Et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة العربي التبسي – تبسة
Université Larbi Tébessi – Tébessa –
Faculté des Sciences et de la Technologie
Département de Génie Électrique

MEMOIRE

Présenté pour l'obtention du **diplôme** de **Master Académique**

En : Filière : Electronique

Spécialité : Instrumentation Électronique

Par : MAINA Imane
MOUMEN Imane

Sujet

Attaques Adverses de l'Apprentissage Profond : Fonctionnement et Menace

Présenté et soutenu publiquement, le 09 / 06 / 2021 devant le jury composé de :

DR AOUICHE Abdel Aziz	MCA	Président
DR KHMAISSIA Sedikke	PHD	Rapporteur
DR HOUAM Lotfi	MCB	Examinateur 1

Promotion : 2020/2021

Dédicaces

Je dédie ce mémoire à mes chers Parents pour leur patience

A ma mère, la lumière de notre vie, qui m'a soutenu et encouragé durant ces années d'études.

A mon très cher père, Qu'Allah te fasse miséricorde Ô papa, et qu'Il te pardonne, et qu'Il te rétribue en bien et Qu'Il te fasse grâce.

A cher grand-mère ma Gratitude ne suffit pas à exprimer cette qu'elle mérite pour tous les sacrifices depuis ma naissance, pendant mon enfance et même à l'âge adulte.

A tous les moments d'enfance passés avec toi ma sœur DOUAA, en gage de ma profonde estime pour l'aide que tu m'as apporté. Tu m'as soutenu, réconforté et encouragé. Puissent nos liens fraternels se consolider et se pérenniser encore plus.

A ma binôme Iman pour tous les souvenirs pendant les années d'études ensemble tu as une place dans mon soeur

A tous mes ami (e) qui m'ont soutenu dans l'accomplissement de cet humble travail A tous mes professeurs et à tous ceux qui se sont engagés dans ces modestes travaux.

A tout ma famille.

Maïna Imen

Dédicaces

A mon très cher père

Pour m'avoir soutenu moralement et matériellement jusqu'à ce jour, pour son amour, et ses encouragements. Que ce travail, soit pour vous, un faible témoignage de ma Profonde affection et tendresse.
Qu'ALLAH le tout puissant te préserve, t'accorde Santé, bonheur et te protège de tout mal.

A ma très chère mère

Autant de phrases aussi expressives soient-elles ne sauraient montrer le degré d'amour et d'affection que j'éprouve pour toi. Tu m'as comblé avec ta tendresse et affection tout au long de mon parcours. Tu n'as cessé de me soutenir et de m'encourager durant toutes les années de mes études. Qu'ALLAH te protège et te donne la santé, le bonheur et longue vie.
A mes frères Anoir, Haroun et Khaled que j'aime tant pour leur petit mot et leur soutie.

A ma binôme Iman pour tous les souvenirs pendant les années d'études ensemble tu as une place dans mon soeur

Tes plus qu'une soeur

Moumen Imen

Remerciements

Tout d'abord, nous remercions Dieu Tout-Puissant de nous avoir donné la volonté et la persévérance d'avoir réalisé ce travail. Nous aimerions profiter de cette occasion pour exprimer notre profonde gratitude à notre encadreur de mémoire ***Dr. KHEMAISSIA SEDDIK*** pour ses précieuses suggestions et ses conseils tout au long de notre période de recherche.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leur précieuses suggestions.

Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Nous remercions tous nos enseignants pour leur dévouement, leur patience et leur contribution à notre formation.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours encouragés au cours de la réalisation de ce mémoire. Merci à tous et à toutes.

Liste de tableaux :

Tableau 1. 1: Les architectures de CNN. 34

Liste des figures:

Figure1. 1: Les réseaux de neurones profonds sont composés de plusieurs couches empilées de neurones artificiels..... 18
Figure1. 2: Comparaison entre Machine Learning et Deep Learning..... 19
Figure1. 3: Neurone biologique. 21
Figure1. 4: Modèle simplifie d'un réseau de neurones artificiels. 22
Figure1. 5: Réseau de neurones artificiels classer un chat. 23
Figure1. 6: Correspondance entre neurone artificiel et neurone biologique. 23
Figure1. 7: La géométrie du neurone artificiel. 25
Figure1. 8: Fonctions d'activation les plus courantes dans les ANN. 25
Figure1. 9: Réseaux de perceptron. 26
Figure1. 10: Réseaux de perceptron Multi Couches. 27
Figure1. 11: Représentation simplifiée d'un RNN simple, avec une couche récurrente et une couche dense. 28
Figure1. 12: Un cheval. 28
Figure1. 13: Image convertie en une matrice de valeurs de pixels. 29
Figure1. 14: l'image réelle (l'image d'entrée) et l'image convoluée. 30
Figure1. 15: La couche entièrement connectée. 32
Figure1. 16: Les étapes de La couche entièrement connectée. 33
Figure1. 17:L'architecture d'un CNN.[19] 33

Figure2. 1: Une démonstration de la génération rapide d'exemple FGSM de confrontation appliquée sur image. 41
Figure 2. 2: Comparaison des images résultant d'une perturbation antagoniste selon la méthode FGSM. 42
Figure2. 3: Exemple de perturbation de la classification d'une série temporelle d'entrée à partir du jeu de données TwoLeadECG en ajoutant un bruit imperceptible calculé à l'aide de la méthode des signes de gradient rapide (FGSM). 43
Figure 2. 4: Un simple autocollant pour tromper l'intelligence artificielle de la voiture autonome. 44
Figure 2.5: Autocollant pour tromper la voiture autonome pensé que la limite de vitesse était de 85 mi / h. 44
Figure 2.6: Le panneau d'arrêt est classé à tort comme panneau de limitation de vitesse. 45
Figure2. 7: Voiture détecter 30 km/h sur autoroute ou la réelle limite est de 120 km/h. 45
Figure2. 8: Les montures de lunettes utilisées pour éviter la reconnaissance. 46
Figure2. 9: Reese Witherspoon (à gauche) portant des lunettes (au centre) imitant Russell Crowe (à droite). 46

Figure2. 10: Un masque 3D élaboré pour tromper le Face ID de l'iPhone X. 46
Figure2. 11: Diagramme d'attaque.[21] 47

Figure 3. 1: Le logo de python. 50
Figure 3. 2: KD nuggets Analytics / Data Science 2020. Sondage sur les logiciels : les principaux outils en 2020 et leur part dans les sondages 2018. 50
Figure 3. 3: Comparaison entre réseau neurone CNN(A) et Resnet50(B). 53
Figure 3. 4: diagramme d'un modèle du programme classification. 54
Figure 3. 5: Diagramme d'implémentation d'un modèle d'attaques contradictoires. 58
Figure 3. 6: Précision de l'entraînement et précision des tests. 65
Figure 3. 7: Matrices de confusion normalisées ResNet50. 66
Figure 3. 8: Matrices de confusion non-normalisées ResNet50. 66
Figure 3. 9: Un graphique qui représente la stratégie de défense. 68

Liste d’abréviation :

ANN: Artificiel Neural Networks 10
API: Application Programming Interface 42
CNN: Les réseaux de neurones convolutionnels 20
DL: Deep Learning 5
ENISA: l'Agence de la cybersécurité de l'Union européenne 33
FC: fully connected layer 24
FGSM: The Fast Gradient Sign Method 31
GPU: Processeur graphique 12
IA: Intelligence Artificielle 5
ML: machine learning 9
MLP: Multi Layer Perceptron 18
NTIC: Technologies de l'information et de la communication 12
PMC: Le perceptron Multi Couches 18
Py: Python 40
RAM: Random Access Memories 40
Relu: Couche unité rectifié linéaire 23
ReLU: Rectified Linear Activation 17
RNN: Réseau de neurones récurrents 19
RVB: Couleur rouge, vert et bleu 21
USD: United State Dollar 36
w:abréviation de poid en anglais (wight) 16

Résumé :

Le Deep Learning (DL) est au cœur de l'essor actuel de l'intelligence artificielle. Il est partout, allant de la prévision du trafic au diagnostic médical, en passant par la conduite autonome. Cependant, la vulnérabilité de sécurité des algorithmes DL aux attaques contradictoires sous la forme de perturbations subtiles des entrées qui conduisent un modèle à prédire des sorties incorrectes a été largement reconnue. Pour les images, de telles perturbations sont souvent trop petites pour être perceptibles, mais elles trompent complètement les modèles d'apprentissage en profondeur. Les attaques adverses constituent une menace sérieuse pour le succès de l'apprentissage en profondeur dans les problèmes du monde réel. Par exemple, en plaçant quelques petits autocollants au sol, les chercheurs ont montré qu'ils pouvaient amener une voiture autonome à se déplacer dans la voie de circulation opposée. D'autres études ont montré qu'apporter des modifications imperceptibles à une image peut amener un système d'analyse médicale à classer un grain de beauté bénin comme malin, et que des morceaux de ruban adhésif peuvent tromper un système de vision par ordinateur en classant à tort un panneau d'arrêt comme un panneau de limitation de vitesse.
Par conséquent, les techniques d'attaque et de défense contradictoires ont attiré de plus en plus l'attention des communautés de l'apprentissage automatique et de la sécurité, et sont devenues un sujet de recherche brûlant ces dernières années. Dans ce mémoire, nous présentons d'abord les fondements théoriques, les algorithmes et les applications des techniques d'attaque contradictoire. Dans le cadre de la taxonomie, les applications des exemples contradictoires sont étudiées. Nous décrivons ensuite quelques efforts de recherche sur les techniques de défense, qui couvrent la large frontière du domaine.

Mots-clés : Machine Learning ─Réseau de neurones profonds ─Vulnérabilité─Adversaire attaque ─Défense contradictoire.

الملخص:

يقع التعلم العميق (DL) في قلب الطفرة الحالية في الذكاء الاصطناعي. إنها في كل مكان، من التنبؤ بحركة المرور إلى التشخيصات الطبية إلى القيادة الذاتية. ومع ذلك، فقد تم الاعتراف على نطاق واسع بالضعف الأمني لخوارزميات DL لهجمات الخصومة في شكل اضطرابات إدخال خفية تقود نموذجًا للتنبؤ بمخرجات غير صحيحة.

بالنسبة للصور، غالبًا ما تكون هذه الاضطرابات صغيرة جدًا بحيث لا يمكن ملاحظتها، لكنها تضلل تمامًا نماذج التعلم العميق. تشكل الهجمات الضارة تهديدًا خطيرًا لنجاح التعلم العميق في مشاكل العالم الحقيقي. على سبيل المثال، من خلال وضع بعض الملصقات الصغيرة على الأرض، أظهر الباحثون أنه يمكنهم الحصول على سيارة ذاتية القيادة تتحرك في المسار المعاكس لحركة المرور. أظهرت دراسات أخرى أن إجراء تغييرات غير محسوسة على صورة ما يمكن أن يتسبب في قيام نظام تحليل طبي بتصنيف الشامة على أنها حميدة على أنها خبيثة، وأن قطع الشريط اللاصق يمكن أن تخدع نظام رؤية الكمبيوتر لتصنيفه عن طريق الخطأ علامة توقف كإشارة حد للسرعة.

نتيجة لذلك، اكتسبت تقنيات الهجوم والدفاع المتضاربة اهتمامًا متزايدًا في مجتمعات التعلم الآلي والأمن، وأصبحت موضوعًا ساخنًا للبحث في السنوات الأخير

في هذه الرسالة، نقدم أولاً الأسس النظرية والخوارزميات وتطبيقات تقنيات الهجوم العدائي. في إطار التصنيف، يتم دراسة تطبيقات الأمثلة المتناقضة. ثم نصف بعض الجهود البحثية حول تقنيات الدفاع، والتي تغطي الحدود الواسعة للمجال.

الكلمات المفتاحية: التعلم الآلي – شبكة عصبية عميقة –الضعف –هجوم الخضم – دفاع متناقض.

Abstract:

Deep learning (DL) is at the heart of the current boom in artificial intelligence. They are everywhere, from traffic forecasts to medical diagnoses to autonomous driving. However, the security vulnerabilities of DL algorithms for adversarial attacks have been widely recognized in the form of hidden input disruptions resulting in an incorrect output prediction model. For images, these disturbances are often too small to notice, but they completely mislead deep learning models. Malicious attacks pose a serious threat to the success of deep learning in real world problems. For example, by placing small stickers on the ground, researchers have shown that they can drive an autonomous car into the opposite traffic lane. Other studies have shown that making subtle changes to an image can cause a medical analysis system to classify a mole as benign as malignant, and that cutting tape can trick a computer vision system into classifying by error a stop sign like a speed limit sign.

As a result, conflicting attack and defense technologies have sparked increasing interest in the machine learning and security communities, and have become a hot topic of research in recent years.
In this thesis, we first present the theoretical foundations, algorithms and applications of contradictory attack techniques. Within the framework of the classification, the applications of contrasting examples are studied. Next, we describe some defense technology research efforts, covering the broad boundaries of the field.

Keywords: Machine learning ─Deep neural network –Vulnerability─Adversary attack ─ Contradictory defense.

Dédicace.....I

Dédicace..... II

Remerciement.....III

Liste de Tableau et Figures IV

Liste d'abréviation V

Résumé VI

Table des matières

INTRODUCTION GENERALE : 13

Chapitre I : Généralité sur l'apprentissage profond et les réseaux de neurones artificiels

1. Introduction 17
2. Deep Learning : 17
3. Le concept du profond : 17
4. Les avantages de l'apprentissage profond : 18
5. Historique : 19
6. Domaines d'application de l'apprentissage profond : 20
7. Les neurones : 20
   - 7.1. Le neurone biologique : 20
   - 7.2. Les réseaux de neurones artificiels (ANN) : 21
8. Comportement de neurones artificiels : 23
9. Quelques types de réseaux de neurones 25
   - 9.1. Perceptron à une seule couche (monocouche): 26
   - 9.2. Perceptron Multi Couches : 26
   - 9.3. Réseau de neurones récurrents : 27
   - 9.4. Les réseaux de neurones convolutionnels (CNN) : 28
     - 9.4.1. Couches convolutives : 29
     - 9.4.2. Couche unité rectifié linéaire (Relu) : 31
     - 9.4.3. Couche Pooling : 31
     - 9.4.4. La couche entièrement connectée : 32
     - 9.4.5. Couche de perte (LOSS) : 32
10. Les architectures de CNN : 34
11. Conclusion : 34

Chapitre II: L'attaque adverse de l'apprentissage profond

1. Introduction : 36
2. Attaque adverse : 36
3. Terminologie associée aux adversaires exemples : 37
3.1. Attaques d'évasion : 37
3.2. Attaques d'empoisonnement : 37
3.3. Attaque non ciblée : 37
2.4. Attaque ciblée : 37
4. Quantité d'information nécessaires : 38
4.1. Attaque boîte noire (Black Box Attack) : 38
4.2. Attaque boîte grise (Gray Box Attack) : 38
4.3. Attaque boîte blanche (White Box Attack) : 38
5. Les menace graves d'intelligence artificielle : 38
6. Attaque adverse avec FGSM (The Fast Gradient Sign Method) : 39
6.1. La descente de gradient : 40
6.2. Exemple algorithme de classification basé sur des séries chronologiques : 43
6.3. Adversaires exemples et la voiture autonome : 43
6.4. Adversaires exemples et la reconnaissance faciale : 45
7. Diagramme d'attaque : 47
8. Conclusion : 47

Chapitre III : Choix technique et résultat expérimentaux

1. Introduction : 49
2. Matériel : 49
3. Choix techniques : 49
3.1. Python (Py) : 49
4. Présentation de l'application : 52
5. Structure de programmation : 53
6. Représentation du système de classification : 54
6.1. Acquisition : 55
6.2. La segmentation d'image : 55
6.3. Extraction des caractéristiques : 55
6.4. Phase de l'entraînement : 55
6.5. Les prédictions : 55

6.6. Classification : 55
6.7. Affichage : 55
7. Résultat : 56
8. Représentation du système d'attaque contradictoire : 58
8.1. Créer adversaire 59
8.2. Phase de l'entraînement : 59
8.3. Prédiction : 60
8.4. Classification : 60
8.5. Affichage : 60
9. Résultat: 60
10. Résultats obtenus et discussion : 64
4. Les défenses adverses : 67
4.1. Les défenses adverses : 67
4.2. Terminologie associée aux La défense adverse 67
5. Stratégies de défense : 68
Conclusion générale : 69

INTRODUCTION GENERALE :

Depuis quelques années, l'Intelligence Artificielle (IA) fait l'objet d'une médiatisation et d'une attention sans précédent. Ce fort regain d'intérêt pour l'IA est notamment lié à d'importantes avancées technologiques qui ont permis d'accroître de façon considérable les performances des ordinateurs dans de nombreux domaines comme la reconnaissance automatique de la parole ou la vision par ordinateur. Ces avancées ont ouvert de vastes perspectives d'introduction de l'IA sous différentes formes (applications, robots, chatbots, etc.)[1][2]

Propulsées par la disponibilité accrue de la puissance de calcul, une connectivité améliorée, et les mégadonnées, les applications de l'IA offrent de fascinantes possibilités de promouvoir la croissance économique et de s'attaquer à un large éventail de problèmes de longue date dans les pays du monde.

L'intelligence artificielle peut favoriser les percées scientifiques, améliorer les diagnostics médicaux, accroître la productivité agricole, aussi elle permet maintenant aux voitures de rouler sans conducteurs, aux robots de devenir de plus en plus autonomes.

Les champs d'application de l'IA dans ces secteurs ne cessent de se multiplier (automatisation de tâches, relation client, logistique, analyse prédictive, diagnostic, analyse de grandes bases de données, etc.). Dans ce cadre, l'IA est le plus souvent vue comme un ensemble de technologies pouvant produire de nombreux bénéfices, notamment en termes de performance (optimisation de processus internes, rapidité d'exécution de tâches, accroissement de la productivité, etc.) et parfois en termes de facilitation du travail voire de réduction de la pénibilité en permettant l'automatisation des tâches fastidieuses ou répétitives.

Ces derniers temps, l'apprentissage automatique a été à la pointe des progrès technologiques. Il semble que ce soit un candidat sérieux pour être l'outil qui pourrait propulser les capacités et l'efficacité humaines au niveau supérieur.

Bien que l'apprentissage automatique soit le terme couramment utilisé, il s'agit d'un sous-ensemble assez important dans le domaine de l'IA. Le terme DL est utilisé pour désigner une approche d'apprentissage automatique qui vise à imiter dans une certaine mesure le fonctionnement du cerveau humain. Cela permet de donner aux machines le pouvoir d'effectuer certaines tâches que les humains peuvent, telles que la détection d'objets, la classification d'objets et bien plus encore. Les modèles d'apprentissage profond qui sont

utilisés pour y parvenir sont souvent connus sous le nom de réseaux de neurones (car ils essaient de reproduire le fonctionnement des connexions neuronales dans le cerveau).[3]

Après une longue traversée, « l'apprentissage profond », est désormais la méthode phare de l'intelligence artificielle (IA). Toutes les grandes entreprises tech s'y mettent : Google, IBM, Microsoft, Amazon, Adobe, Yandex ou encore Baidu y investissent des fortunes. Facebook également.

Ce système d'apprentissage et de classification, basé sur des « réseaux de neurones artificiels » numériques, est utilisé par Siri, Cortana et Google Now pour comprendre la voix, être capable d'apprendre à reconnaître des visages. [4]

Cependant, comme tout autre méthode, les réseaux de neurones ont également leur propre ensemble de vulnérabilités.

Récemment, la vulnérabilité de sécurité des algorithmes DL aux échantillons contradictoires a été largement reconnue. Les échantillons contradictoires peuvent conduire vers comportements catastrophique des modèles DL tout en étant perçus comme bénins par les humains. Ces derniers temps, la vulnérabilité qui a pris le plus de place est connue sous le nom d'exemples contradictoires.

Cette mémoire vise à faire la lumière sur la nature des exemples contradictoires et certaines des appréhensions et problèmes qui surviennent avec le développement de méthodes d'apprentissage en profondeur en raison de ces vulnérabilités.

Structure du mémoire :

Chapitre I : “Généralité sur l’apprentissage profond et les réseaux de neurones artificiels“

Le premier chapitre décrit les notions de base pour l'apprentissage profond et généralité sur les neurones artificielles et différentes méthodes de classification des images, et les algorithmes d’apprentissage des réseaux de neurones.

Chapitre II: ” L’attaque adverse de l’apprentissage profond ”

Dans ce chapitre, nous avons présenté un aperçu de la récente attaque contradictoire représentative. Nous avons étudié les idées et les méthodologies des méthodes et algorithmes proposés. Certains problèmes fondamentaux, tels que la causalité d'échantillons contradictoires et l'existence d'une frontière générale robuste, ont également été étudiés.

Chapitre III : “Choix technique et résultat expérimentaux “

Ce chapitre 3 illustre l’implémentation et l’expérimentation de notre système, les outils et logiciels que nous avons eu à utiliser pour le développement, et la réalisation de notre système. Et enfin, nous terminerons ce mémoire par une conclusion générale.

CHAPITRE I :
GENERALTÉ SUR L'APPRENTISSAGE PROFOND ET LES RESEAUX DE NEURONES ARTIFICIELS.

1. Introduction :

Dans ce chapitre, nous allons comprendre ce qu'est le Deep Learning, ce que sont les réseaux de neurones et comment utiliser ces réseaux convolutifs pour classer les images. Mais avant de discuter de tout cela, mentionnons d'abord quelques concepts liés à l'apprentissage en profondeur.

2. Deep Learning :

La notion d'apprentissage profond est tout d'abord une traduction directe du terme anglais

« Deep Learning » c'est un type de machine Learning (ML) (apprentissage automatique) où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées.[5]

Dans l'apprentissage automatique, un programme analyse un ensemble de données afin de tirer des règles qui permettront de tirer des conclusions sur de nouvelles données .

Le DL s'appuie sur un réseau de neurones artificiels s'inspirant du fonctionnement des neurones biologiques du cerveau humain. Ce réseau est composé de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente. [6][7]

Le Deep Learning a la capacité d'extraire des caractéristiques à partir des données brutes grâce aux multiples couches de traitement composé de multiples transformations linéaires et non linéaires et apprendre sur ces caractéristiques petit à petit à travers chaque couche avec une intervention humaine minime

Par exemple : pour la reconnaissance visuelle, des premières couches d'unités identifient des lignes, des courbes, des angles… des couches supérieures identifient des formes, des combinaisons de formes, des objets, des contextes…

Les progrès de l'apprentissage profond ont été possibles notamment grâce à l'augmentation de La puissance des ordinateurs et au développement de grandes bases de données (« big data »).[8]

3. Le concept du profond :

Un réseau de neurones est un modèle de cerveau ou une machine virtuelle composée de milliers d'unités (neurones) qui effectuent des calculs. Plus précisément, les unités logiques et décisionnelles (neurones, perceptrons) relient les données d'entrée et de sortie à travers des réseaux complexes (réseaux, cerveaux) capables de prendre des décisions complexes.[9]

A l'origine, ces systèmes portaient le nom de réseaux artificiels de neurones (ANN, Artificiel Neural Networks) afin de les différencier des systèmes biologiques. Ils se composent en général d'un certain nombre de données d'entrées et de sorties (input / output layer), d'un réseau étroit de neurones et de plusieurs strates intermédiaires (hidden layers). Ces couches intermédiaires permettent de traiter des problèmes complexes ; sans elles, le système ne résout que des calculs simples. Le nombre de couches est donc un facteur décisif pour la complexité du système, et de l'apprentissage ; les données s'associent d'une couche à l'autre, les résultats d'une première couche servant d'entrée à la prochaine, et ainsi de suite afin d'aboutir à une prise de décision complexe. Ce fonctionnement en strates donne toute sa profondeur au réseau et à l'apprentissage. L'adjectif « profond » s'entend ici dans tous les sens du terme.

L'exemple ci-dessus présente des unités d'entrées, des sortie et deux couches intermédiaires (couches cachées). Vous constatez que les neurones sont « fortement interconnectés », ce qui constitue une propriété essentielle des réseaux de neurones. C'est justement ce qui autorise les relations, fonctions ou décisions d'être complexes ; sans cette propriété, les relations entrée-sortie seraient relativement simples.

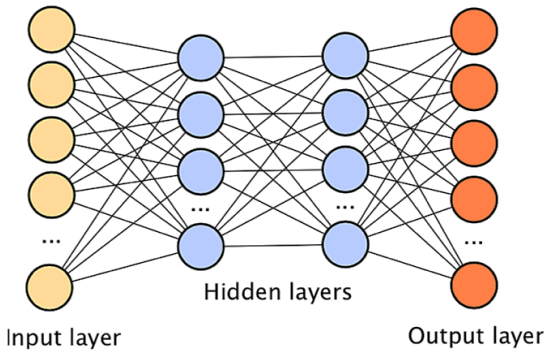


Figure1. 1: Les réseaux de neurones profonds sont composés de plusieurs couches empilées de neurones artificiels.

L'objectif n'est pas de créer un modèle exact du cerveau, mais bien de reproduire sa capacité d'apprentissage et de reconnaissance de connexions complexes. Un être humain peut avoir jusqu'à 100 milliards de neurones qui fonctionnent à une fréquence d'environ 1 kHz ; un processeur moderne fonctionne avec 2 milliards de transistors à 3 GHz.[9]

4. Les avantages de l'apprentissage profond :

les différents algorithmes du DL ne sont apparus qu'à l'échec de l'apprentissage automatique tentant de résoudre une grande variété de problèmes de l'intelligence artificielle (l'IA)

- Afin d'améliorer le développement des algorithmes traditionnels dans de telles tâches de l'IA.
- De développer une grande quantité de données telle que les big data.
- De s'adapter à n'importe quel type de problème
- D'extraire les caractéristiques de façon automatique[10][11]

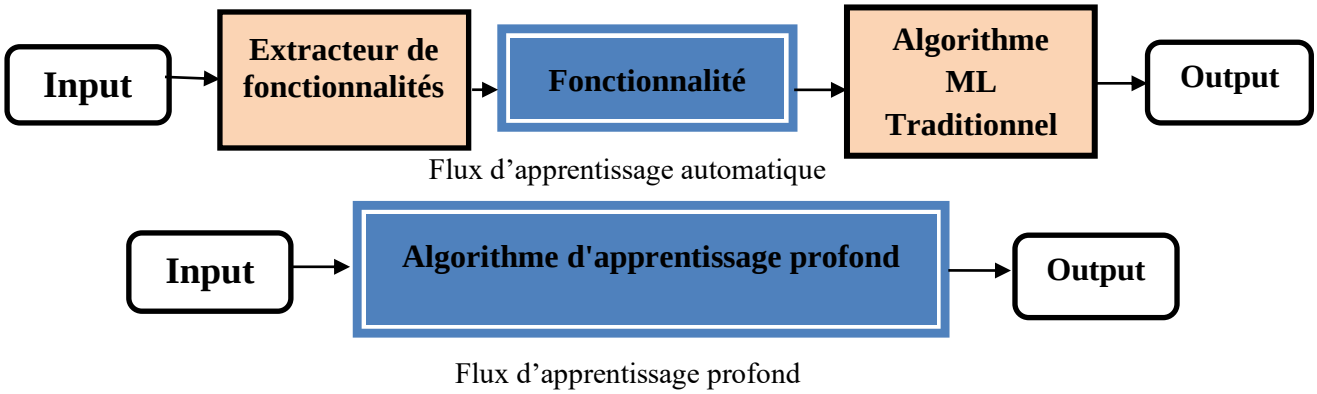


Figure1. 2: Comparaison entre Machine Learning et Deep Learning.

5. Historique :

L'idée du Deep Learning n'est pas une idée récente, mais elle date en réalité des années 1980, la première fois que l'on parle de Deep Learning, c'est grâce à la professeure Rina Dechter en 1986. Ensuite, cette approche est mise en pratique par Yann LeCun en 1989. L'actuel boss de l'IA chez Facebook avait à l'époque utilisé un réseau de neurones artificiel profond afin de reconnaître les codes postaux écrits à la main sur des lettres. Un programme simple aujourd'hui, mais qui avait nécessité trois jours d'apprentissage à l'époque.

En collaboration Yann LeCun avec deux autres informaticiens, Kunihiko Fukushima et Geoffrey Hinton, ils mettent au point un type d'algorithme particulier appelé Convolutional neural network.

C'est dans ce contexte qu'en 2007 le STANFORD VISION LAB, avec Fei-Fei Li à sa tête, développent un agrégateur d'images où sont consignés et étiquetés quelques millions de photos : ImageNet. En 2010, ImageNet regroupe 15 000 000 d'images toutes catégorisées en fonction de leurs caractéristiques propres (véhicules, animaux, ...).

Des recherches et des études sur la structure des réseaux de neurones continueront d'animer la communauté scientifique jusqu'en 2009, année où cette pratique prend son envol. On considère cette année comme le big bang du DL. À ce moment-là, Nvidia met ses processeurs graphiques (GPU) à contribution. Pour Andrew Ng, cofondateur de Google Brain,

l'exploitation des GPU pourrait créer des systèmes d'apprentissage profond jusqu'à 100 fois plus rapides. L'entraînement des algorithmes passerait de plusieurs semaines à seulement quelques jours.

C'est ainsi que Andrew Ng crée une architecture avec bien plus de couches et de neurones qu'auparavant. Il l'entraîne ensuite avec des contenus provenant de 10 millions de vidéos YouTube afin que son programme puisse identifier et extraire les images avec des chats. Nous sommes en 2012.[12][13]

6. Domaines d'application de l'apprentissage profond :

Ces techniques se développent dans le domaine de l'informatique appliquée aux NTIC (reconnaissance visuelle — par exemple d'un panneau de signalisation par un robot ou une voiture autonome et vocale notamment) à la robotique, à la bio-informatique, la reconnaissance ou comparaison de formes, la sécurité, la santé, etc…

L'apprentissage profond peut par exemple permettre à un ordinateur de mieux reconnaitre des objets hautement déformables et/ou analyser par exemple les émotions révélées par un visage photographié ou filmé, ou analyser les mouvements et position des doigts d'une main, ce qui peut être utile B pour traduire le langage des signes, améliorer le positionnement automatique d'une caméra, etc…

Elles sont utilisées pour certaines formes d'aide au diagnostic médical (ex. : reconnaissance automatique d'un cancer en imagerie médicale).

Ou de prospective ou de prédiction (ex. : prédiction des propriétés d'un sol filmé par un robot).

7. Les neurones :

Les réseaux de neurones artificiels constituent l'une des approches IA dont le développement se fait à travers les méthodes par lesquelles l'homme essaye toujours d'imiter la nature et de reproduire des modes de raisonnement et de comportement qui lui sont propre.

Nous présentons dans cette section de chapitre un état de l'art de ces réseaux de neurones.

7.1. Le neurone biologique :

Le neurone biologique (voir figure 1.3) est une cellule nerveuse qui constitue l'unité fonctionnelle fondamentale du système nerveux de tous les animaux. Les neurones existent pour communiquer les uns avec les autres et transmettre des impulsions électrochimiques à travers les synapses, d'une cellule à l'autre, à condition que l'impulsion soit suffisamment puissante pour activer la libération de produits chimiques à travers une fente synaptique. La

force de l'impulsion doit dépasser un seuil minimal, sinon les produits chimiques ne seront pas libérés.

La figure présente les principales parties de la cellule nerveuse : le soma, les dendrites, les axones et les synapses.

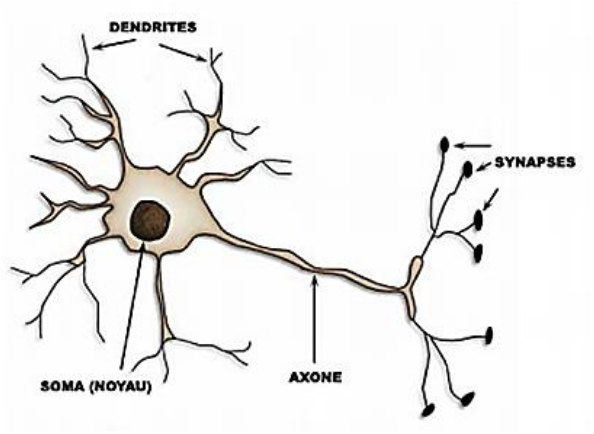


Figure1. 3: Neurone biologique.

7.2. Les réseaux de neurones artificiels (ANN) :

Au sein d'un processeur, l'unité logique se compose de transistors ; on pourrait également y trouver un réseau neuronal « câblé », mais il faudrait qu'il soit « adaptatif », qu'il ait une « capacité d'apprentissage ». En effet, la réponse d'un neurone à des impulsions entrantes doit pouvoir évoluer tout au long du processus d'apprentissage. C'est ce qu'on appelle la « pondération » : un neurone évalue (pondère) diverses variables d'entrée pour obtenir la variable de sortie souhaitée. C'est pourquoi les neurones sont généralement des fonctions mathématiques qui relient entre elles des variables d'entrée et de sortie.

Dans la phase d'apprentissage, les neurones modifient leur comportement de pondération et affinent les résultats de sortie en fonction des variables d'entrée. Il doit donc y avoir un retour d'information du résultat global qui influence chaque neurone. On peut donc dire que les variables d'entrée et de sortie d'un réseau neuronal sont connues, mais que les valeurs des neurones, surtout dans les couches cachées restent inconnues.

Pris isolément, un réseau neuronal qui n'a pas été entraîné ne « connaît » rien et fournit des résultats aléatoires, voire chaotiques pour l'utilisateur. Seul un système entraîné fournira le résultat souhaité. Si le problème posé est simple, un programme simple, plus facile à déboguer, pourra le résoudre. Pour un problème plus complexe, on aura recours à un réseau de neurones, que l'on entraînera à l'aide de grands ensembles de données. Chaque neurone

peut fournir des variables de sortie complexes et réagir linéairement ou non linéairement aux variables d'entrée. C'est là un point assez subtil, car les neurones doivent être en mesure de réagir à l'ensemble des possibilités afin de fournir un résultat adéquat. Ceci implique deux choses : soit le programmateur du réseau neuronal connaît toutes les connexions possibles internes au réseau, soit il conçoit le réseau de manière si complexe qu'il couvre « toutes » les possibilités.[10]

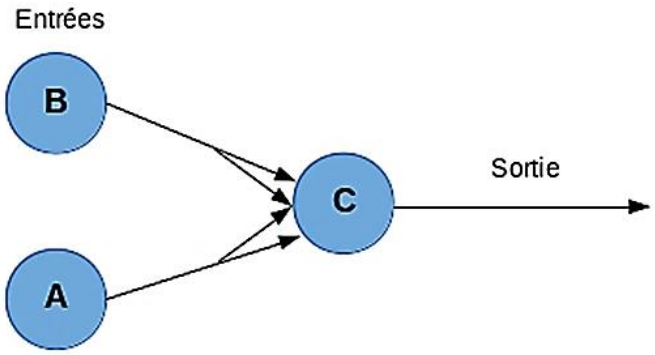


Figure1. 4: Modèle simplifie d'un réseau de neurones artificiels.

Pour comprendre comment fonctionne les réseaux de neurones, prenons un exemple concret de reconnaissance d'images. Imaginons que le réseau de neurones soit utilisé pour reconnaître les photos qui comportent au moins un chat. Pour pouvoir identifier les chats sur les photos, l'algorithme doit être en mesure de distinguer les différents types de chats, et de reconnaître un chat de manière précise quel que soit l'angle sous lequel il est photographié.

Afin d'y parvenir, le réseau de neurones doit être entraîné. Pour ce faire, il est nécessaire de compiler un ensemble d'images d'entraînement pour pratiquer le DL. Cet ensemble va regrouper des milliers de photos de chats différents, mélangés avec des images d'objets qui ne sont pas des chats. Ces images sont ensuite converties en données et transférées sur le réseau. Les ANN assignent ensuite un poids aux différents éléments. La couche finale de neurones va alors rassembler les différentes informations pour déduire s'il s'agit ou non d'un chat.

Pour comprendre comment fonctionne les réseaux de neurones, prenons un exemple concret de reconnaissance d'images. Imaginons que le réseau de neurones soit utilisé pour reconnaître les photos qui comportent au moins un chat. Pour pouvoir identifier les chats sur les photos, l'algorithme doit être en mesure de distinguer les différents types de chats, et de reconnaître un chat de manière précise quel que soit l'angle sous lequel il est photographié.

Afin d'y parvenir, le réseau de neurones doit être entraîné. Pour ce faire, il est nécessaire de compiler un ensemble d'images d'entraînement pour pratiquer le DL. Cet ensemble va regrouper des milliers de photos de chats différents, mélangés avec des images d'objets qui ne sont pas des chats. Ces images sont ensuite converties en données et transférées sur le réseau. Les ANN assignent ensuite un poids aux différents éléments. La couche finale de neurones va alors rassembler les différentes informations pour déduire s'il s'agit ou non d'un chat.

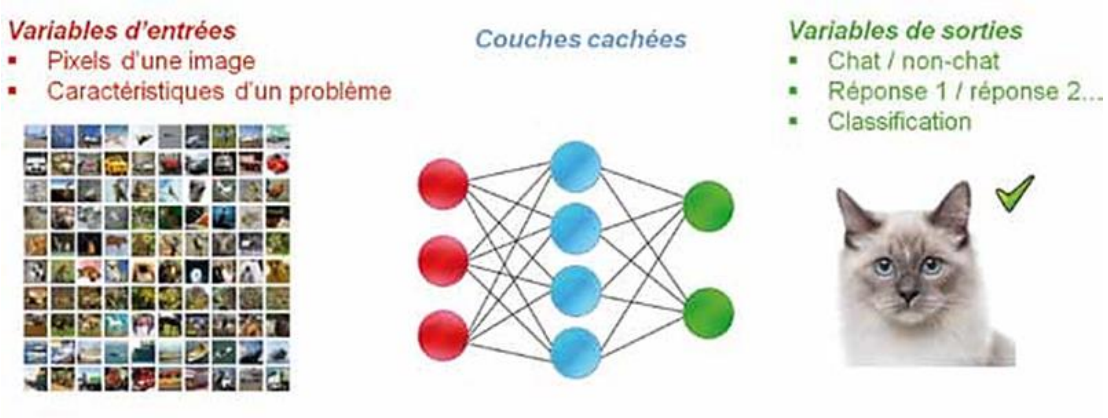


Figure1. 5: Réseau de neurones artificiels classer un chat.

Le réseau de neurones va ensuite comparer cette réponse aux bonnes réponses indiquées par les humains. Si les réponses correspondent, le réseau garde cette réussite en mémoire et s'en servira plus tard pour reconnaître les chats. Dans le cas contraire, le réseau prend note de son erreur et ajuste le poids placé sur les différents neurones pour corriger son erreur. Le processus est répété des milliers de fois jusqu'à ce que le réseau soit capable de reconnaître un chat sur une photo dans toutes les circonstances.[14][15]

8. Comportement de neurones artificiels :

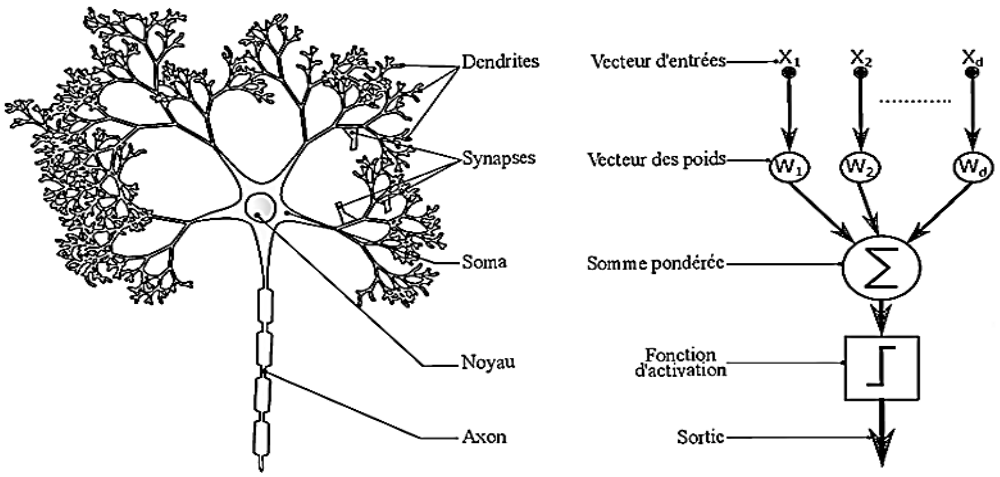


Figure1. 6: Correspondance entre neurone artificiel et neurone biologique.

Par analogie avec la biologie, ANN (figure 1.6), reçoit l'information provenant des entrées i $\boldsymbol{x(i = 1, 2, 3 \ldots n)}$ par l'intermédiaire des connexions dont on affecte à chacune d'elles un poids $\boldsymbol{w_i}$ abréviation de Wight en anglais pondérant l'information, et aussi représentatif de la force de la connexion. Le neurone artificiel fonctionne en deux étapes : La première phase représente les prétraitements des donnés reçus en calculant le potentiel $\boldsymbol{v_j}$des neurones j par la fonction suivante :

$$\boldsymbol{v_j = b_j + \sum_{i=0}^{n} w_{i,j}\, x_i} \qquad \text{(I.1)}$$

Où

$\boldsymbol{w_{i,j}}$: désigne le poids de la connexion liant le neurone $\boldsymbol{j}$ à l'entrée $\boldsymbol{i}$;

$\boldsymbol{b_j}$: terme constant appelé biais, il est considéré comme le poids d'une entrée 0 x égal à 1.

Ainsi la relation s'écrit plus simplement :

$$\boldsymbol{v_{j=} \sum_{i=0}^{n} w_{i,j}\, x_i} \qquad \text{(I.2)}$$

Dans la deuxième phase, une fonction de transfert g appelée également fonction d'activation, calcule la valeur de l'état interne $\boldsymbol{s_j}$ du neurone j à partir de la valeur du potentiel $\boldsymbol{v_j}$. Cette valeur désignera la sortie du neurone :

$$\boldsymbol{s_j = g(v_j) = g(\sum_{i=0}^{n} w_{i,j}\, x_i)} \qquad \text{(I.3)}$$

Le choix de la fonction d'activation se révèle dans certains cas être un élément constitutif important des réseaux de neurones. Ainsi, le neurone peut être défini mathématiquement comme étant une fonction algébrique, non linéaire (suivant g) et bornée, des entrées $\boldsymbol{x_i}$ et paramétrée par les poids $\boldsymbol{w_{i,j}}$ *:*

$$\boldsymbol{S_j = g(x_i, w_{j,i}) = g_{w_{i-j}}(x_i) \qquad i = (1, 2, 3 \ldots n)} \qquad \text{(I.4)}$$

La fonction d'activation est un élément essentiel aux réseaux de neurones. Elle introduit une non-linéarité dans le modèle et permet d'étendre la dimension de l'espace des hypothèses. Sans cette fonction d'activation, le modèle pourrait seulement apprendre des transformations linéaires ce qui restreindrait trop l'espace des hypothèses.[16][7]

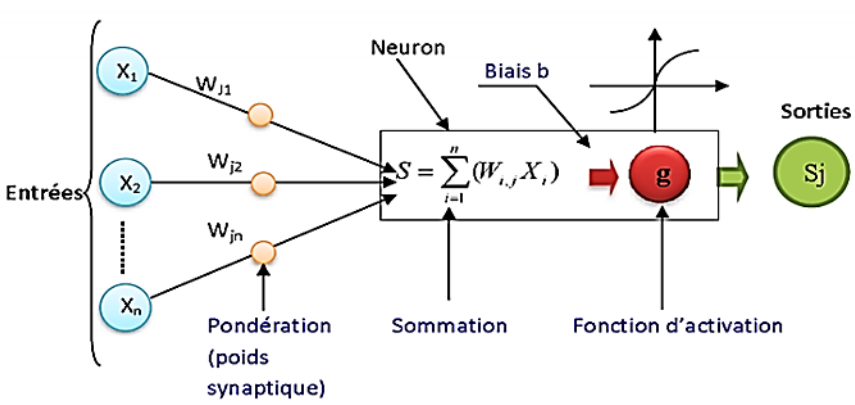


Figure1. 7: La géométrie du neurone artificiel.

Les fonctions d'activation les plus courantes sont :

• La fonction ReLU (Rectified Linear Activation)

• La fonction sigmoïde

La fonction ReLU (figure 1.8) permet d'éviter les valeurs négatives à la sortie du neurone puisqu'elle met à zéro toutes les valeurs négatives alors que les valeurs positives sont inchangées. C'est une des fonctions d'activation les plus utilisées dans les réseaux de neurones. La fonction sigmoïde (figure 1.8) retourne une valeur comprise entre 0 et 1. C'est une fonction d'activation souvent utilisée pour la dernière couche afin d'obtenir à la sortie du réseau un score qui peut s'interpréter comme une probabilité.[6]

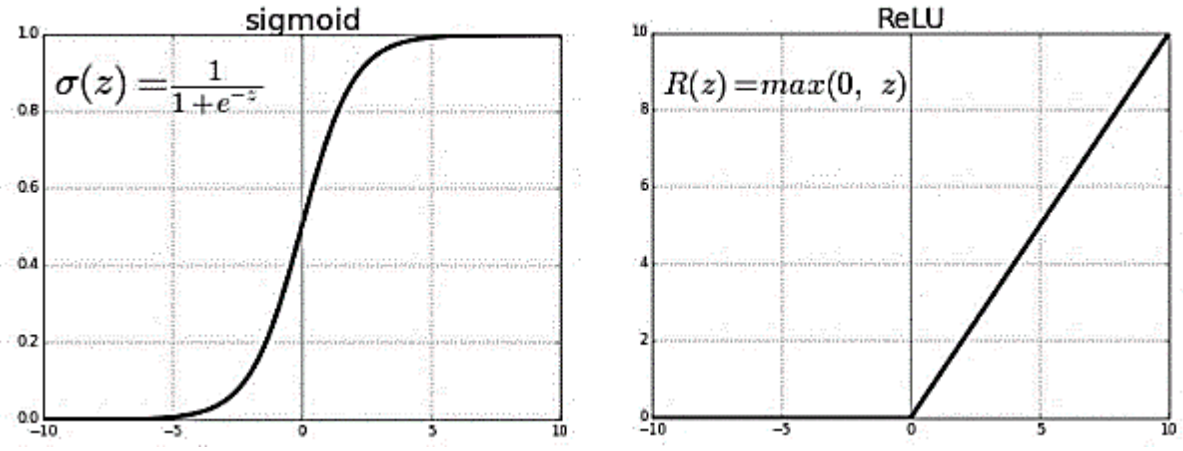


Figure1. 8: Fonctions d'activation les plus courantes dans les ANN.

9. Quelques types de réseaux de neurones :

Il existe beaucoup de types de ANN, chaque type étant développe pour un objectif Particulier.

9.1. Perceptron à une seule couche (monocouche) :

La forme la plus simple d'un réseau de neurone est le perceptron. Ce réseau est considéré parmi les premiers réseaux de neurones. Il a été inventé en 1957 par Rosenblatt.

Le perceptron se compose d'un neurone artificiel à poids ajustable et d'un seuil. Il n'a qu'une seule sortie à laquelle toutes les entrées sont connectées. Les entrées et la sortie sont booléennes.

Dans ce type de réseaux, seulement les poids entre les unités d'entrées et la sortie peuvent être modifiés, tandis que la sortie de neurone ne peut prendre que deux états : -1 et 1 ou 0 et 1.

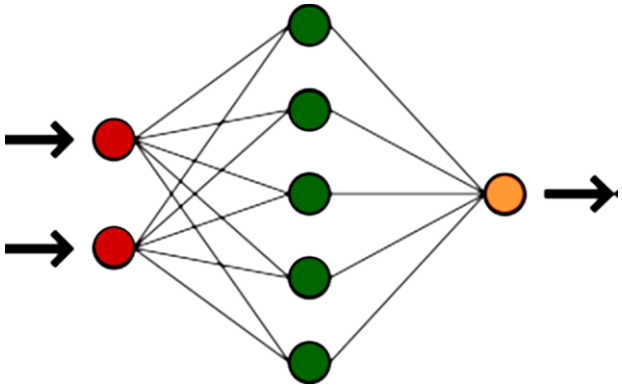


Figure1. 9: Réseaux de perceptron.

9.2. Perceptron Multi Couches :

Le perceptron Multi Couches PMC ou MLP (Multi Layer Perceptron) en anglais est une extension du perceptron monocouche, avec une ou plusieurs couches cachées entre l'entrée et la sortie.

L'idée principale est de grouper des neurones dans une couche. En place ensuite bout à bout plusieurs couches et on connecte complètement les neurones de deux couches adjacentes (Figure 1.10). Les entrées des neurones de la deuxième couche sont donc en fait les sorties des neurones de la première couche. Les neurones de la première couche sont reliés au monde extérieur et reçoivent tous le même vecteur d'entrée. Ils calculent alors leurs sorties qui sont Transmises aux neurones de la deuxième couche, etc. Les sorties des neurones de la dernière couche forment la sortie du réseau. Les réseaux multicouches sont beaucoup plus puissants que les réseaux simples à une seule couche. En utilisant deux couches (une couche cachée et une couche de sortie) à condition d'employer une fonction d'activation sigmoïde sur la couche cachée, on peut entrainer un réseau à produire une approximation de la plupart des fonctions, avec une précision arbitraire (cela peut cependant requérir un grand nombre de

neurones sur la couche cachée). Sauf dans de rares cas, les réseaux de neurones artificiels exploitent deux ou trois couches.

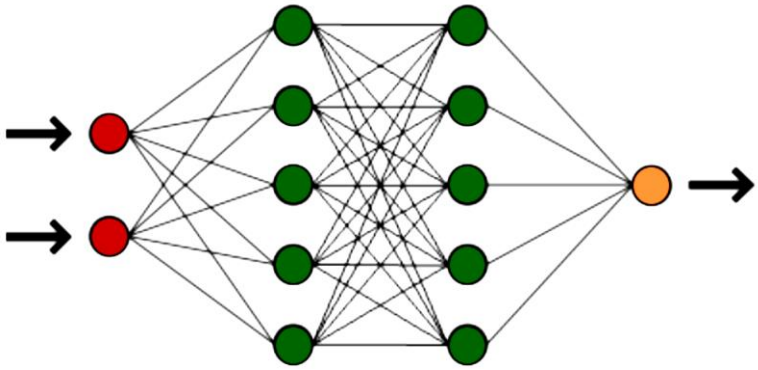


Figure1. 10: Réseaux de perceptron Multi Couches.

9.3. Réseau de neurones récurrents :

L'idée derrière les RNN est d'utiliser des informations séquentielles. Dans un réseau neuronal traditionnel, nous supposons que toutes les entrées (et les sorties) sont indépendantes les unes des autres. Mais pour de nombreuses tâches, c'est une très mauvaise idée. Si on veut prédire le prochain mot dans une phrase, il faut connaître les mots qui sont venus avant. Les RNN sont appelés récurrents, car ils exécutent la même tâche pour chaque élément d'une séquence, la sortie étant dépendante des calculs précédents.

Une autre façon de penser les RNN est qu'ils ont une « mémoire » qui capture l'information sur ce qui a été calculé jusqu'ici. En théorie, les RNN peuvent utiliser des informations dans des séquences arbitrairement longues, mais dans la pratique, on les limite à regarder seulement quelques étapes en arrière. Il est utilisé pour :

- La modélisation du langage et génération de texte.
- La traduction automatique.
- La reconnaissance vocale Et la description des images.[17]

Comme les réseaux de neurones récurrents possèdent de nombreuses connexions, on utilise souvent des représentations simplifiées où une seule flèche représente une matrice de poids W complète. La figure suivante représente ainsi un réseau classique, avec une couche récurrente suivie d'une couche dense classique :

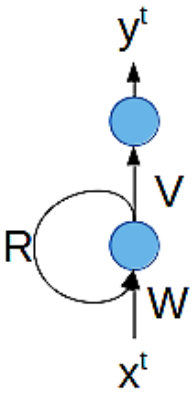


Figure1. 11: Représentation simplifiée d'un RNN simple, avec une couche récurrente et une couche dense.

9.4. Les réseaux de neurones convolutionnels (CNN) :

En apprentissage automatique, un réseau de neurone convolutifs (ou réseau de neurones à convolution, ou CNN ou ConvNet) sont un type de réseau de neurones spécialisés pour le traitement de données ayant une topologie semblable à une grille. Qui se sont avérés très efficaces dans des domaines tels que la reconnaissance et la classification d'images et vidéos. CNN a réussi à identifier les visages, les objets, panneaux de circulation et auto-conduite des voitures. Récemment, les CNN ont été efficaces dans plusieurs tâches de traitement du langage naturel (telles que la classification des phrases).[18]

Figure1. 12: Un cheval.

Lorsque nous transmettons l'image à un ordinateur, il la convertit essentiellement en une matrice de valeurs de pixels. Les valeurs de pixel vont de 0 à 255, et les dimensions de cette matrice seront de [largeur de l'image x hauteur de l'image x nombre de canaux].

Une image en niveaux de gris a un canal et les images colorées ont trois canaux rouge, vert et bleu ***(RVB).***

Comme indiqué dans le diagramme suivant, l'image en niveaux de gris d'entrée sera convertie en une matrice de valeurs de pixels allant de 0 à 255, les valeurs de pixels représentant l'intensité des pixels à ce point :

Figure1. 13: Image convertie en une matrice de valeurs de pixels.

D'accord, nous avons maintenant une matrice d'entrée de valeurs de pixels.

9.4.1. Couches convolutives :

La couche convolutionnelle est la première couche centrale du CNN. C'est l'un des éléments constitutifs d'un CNN et est utilisé pour extraire des caractéristiques importantes de l'image. C'est là que nous utilisons une opération de convolution qui va extraire toutes les caractéristiques importantes de l'image qui caractérisent le cheval. Ainsi, l'opération de convolution nous aide à comprendre en quoi consiste l'image.

Comme nous le savons, chaque image d'entrée est représentée par une matrice de valeurs de pixels. Outre la matrice d'entrée, nous avons également une autre matrice appelée matrice de filtre. La matrice de filtre est également connue sous le nom de noyau, ou simplement de filtre, comme illustré dans le diagramme suivant :

0	13	13
7	7	7
9	11	11

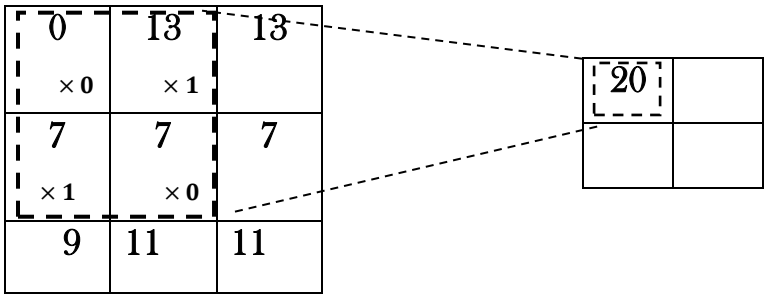
Input matrix

0	1
1	0

Filtre

Nous prenons la matrice de filtre, la glissons sur la matrice d'entrée d'un pixel, effectuons une multiplication par élément, additionnons les résultats et produisons un seul nombre.

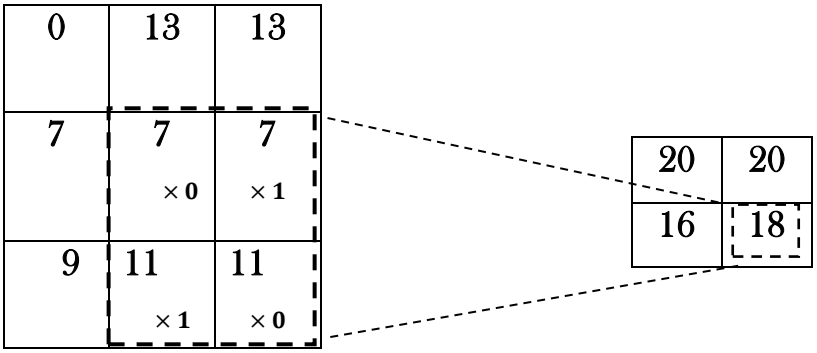
Comprenons mieux cela à l'aide du schéma suivant :



Comme vous pouvez le voir dans le diagramme précédent, nous avons pris la matrice de filtre et l'avons placée au-dessus de la matrice d'entrée, effectué une multiplication élémentaire, additionné leurs résultats et produit le nombre unique. Ceci est démontré comme suit :

$$(\mathbf{0} * \mathbf{0}) + (\mathbf{13} * \mathbf{1}) + (\mathbf{7} * \mathbf{1}) + (\mathbf{7} * \mathbf{0}) = \mathbf{20}$$

Maintenant, encore une fois, nous faisons glisser la matrice de filtre sur la matrice d'entrée d'un pixel et effectuons la même opération, comme indiqué dans le diagramme suivant :



Comme nous l'avons appris, l'opération de convolution est utilisée pour extraire des entités, et la nouvelle matrice, c'est-à-dire les cartes d'entités, représente les entités extraites. Si nous traçons les cartes de caractéristiques, nous pouvons voir les caractéristiques extraites par l'opération de convolution.[19]

Le diagramme suivant montre l'image réelle (l'image d'entrée) et l'image convoluée (la carte des caractéristiques). Nous pouvons voir que notre filtre a détecté les bords de l'image réelle comme une caractéristique :

Figure1. 14: l'image réelle (l'image d'entrée) et l'image convoluée.

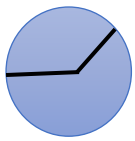
9.4.2. Couche unité rectifié linéaire (Relu) :

Un élément important dans l'ensemble du processus est l'Unité linéaire rectifiée ou Relu. Les mathématiques derrière ce concept sont assez simples encore une fois : chaque fois qu'il y a une valeur négative dans un pixel, on la remplace par un 0. Ainsi, on permet au CNN de rester en bonne santé (mathématiquement parlant) en empêchant les valeurs apprises de rester coincer autour de 0 ou d'exploser vers l'infinie.

C'est un outil fondamental car sans lequel le CNN ne produirait pas vraiment les résultats qu'on lui connaît.

Relu

0	-13	13
7	7	-7
-9	11	11



0	0	13
7	7	0
0	11	11

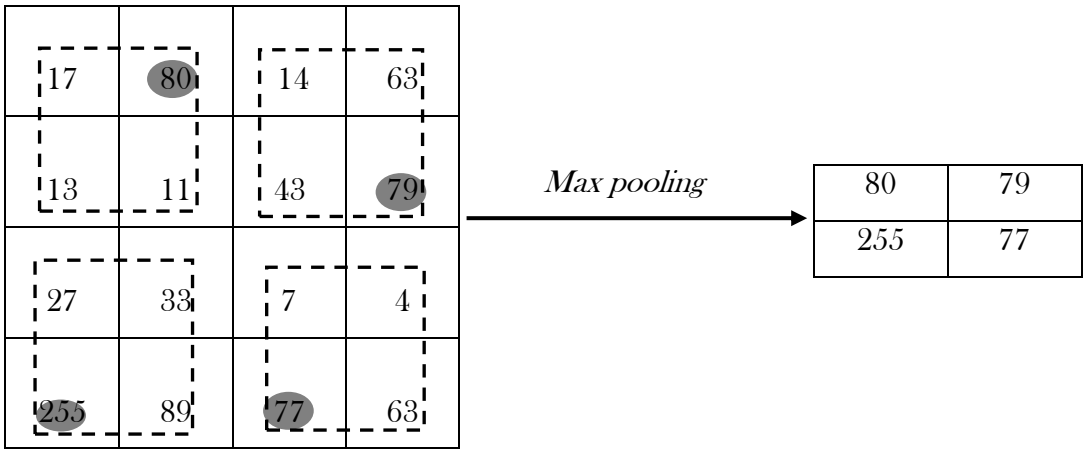
Le résultat d'une couche Relu est de la même taille que ce qui lui est passé en entrée, avec simplement toutes les valeurs négatives éliminées.

Le résultat d'une couche Relu est de la même taille que ce qui lui est passé en entrée, avec simplement toutes les valeurs négatives éliminées.

9.4.3. Couche Pooling :

Un autre outil très puissant utilisé par les CNN s'appelle le Pooling. Le Pooling est une méthode permettant de prendre une large image et d'en réduire la taille tout en préservant les informations les plus importantes qu'elle contient. Les mathématiques derrière la notion de pooling ne sont une nouvelle fois pas très complexe. En effet, il suffit de faire glisser une petite fenêtre pas à pas sur toutes les parties de l'image et de prendre la valeur maximum de cette fenêtre à chaque pas. Après avoir procédé au pooling, l'image n'a plus qu'un quart du nombre de ses pixels de départ parce qu'il garde à chaque pas la valeur maximale contenue dans la fenêtre, il préserve les meilleures caractéristiques de cette image. Cela signifie qu'il ne se préoccupe pas vraiment d'où a été extraite la caractéristique dans l'image.
Le résultat est que le CNN peut trouver si une caractéristique est dans une image, sans se soucier de l'endroit où elle se trouve.
Il existe plusieurs types de pooling :

- Le max pooling.
- Le mean pooling.
- Le sum pooling.[19]



9.4.4. La couche entièrement connectée :

L'ensemble successives des cartes de convolutions, pooling et ReLu fournissent au final un ensemble de caractéristiques données sous forme de carte 2D. Ces cartes sont concaténées en un vecteur de caractéristiques, appelé code CNN. Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée d'une ou de plusieurs couches entièrement connectées (fullyconnected ou FC) qu'on peut assimiler au perceptron multicouche. Les neurones dans une couche entièrement connectée ont des connexions vers les sorties de la couche précédente.

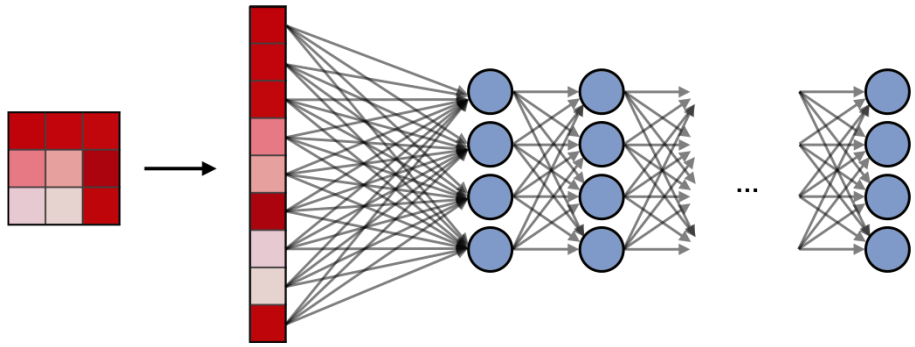


Figure1. 15: La couche entièrement connectée.

9.4.5. Couche de perte (LOSS) :

La couche de perte suit la couche entièrement connectée et gère les ajustements de poids sur le réseau. Avant que l'entrainement du réseau ne commence, les poids des couches convolution et entièrement connectées reçoivent des valeurs aléatoires. Ensuite, pendant l'entraînement, la couche de perte vérifie en permanence les suppositions de la couche

entièrement connectée par rapport aux valeurs réelles dans le but de minimiser autant que possible la différence entre l'estimation et la valeur réelle. La couche de perte effectue cela en ajustant les poids à la fois dans la couche de convolution et dans les couches entièrement connectées.

Maintenant, nous devons classer ces fonctionnalités extraites. Nous avons donc besoin d'un algorithme capable de classer ces entités extraites et de nous dire si les entités extraites sont les caractéristiques d'un cheval, ou autre chose. Afin de faire cette classification, nous utilisons un réseau de neurones feedforward.

Nous aplatissons la carte des caractéristiques et la convertissons en vecteur, et la fournissons en tant qu'entrée au réseau anticipé. Le réseau à anticipation prend cette carte de caractéristiques aplatie comme entrée, applique une fonction d'activation, telle que sigmoïde, et renvoie la sortie, indiquant si l'image contient un cheval ou non ; cela s'appelle une couche entièrement connectée et est illustré dans le diagramme suivant :

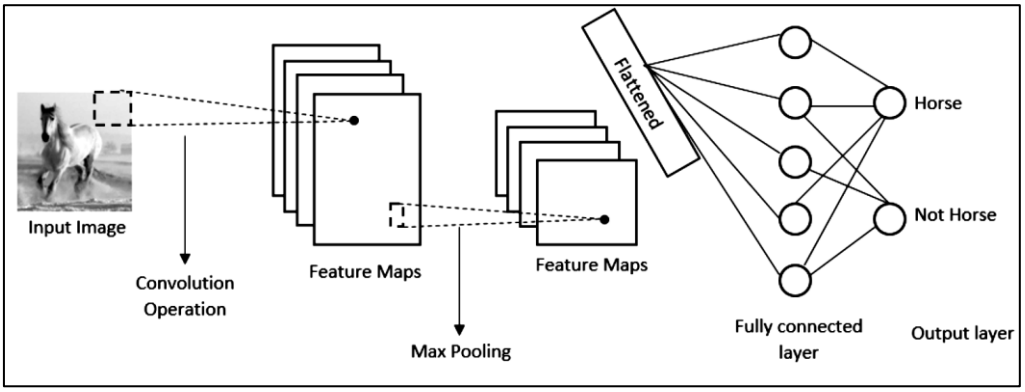


Figure1. 16: Les étapes de La couche entièrement connectée.

L'architecture d'un CNN est illustrée dans le schéma suivant :

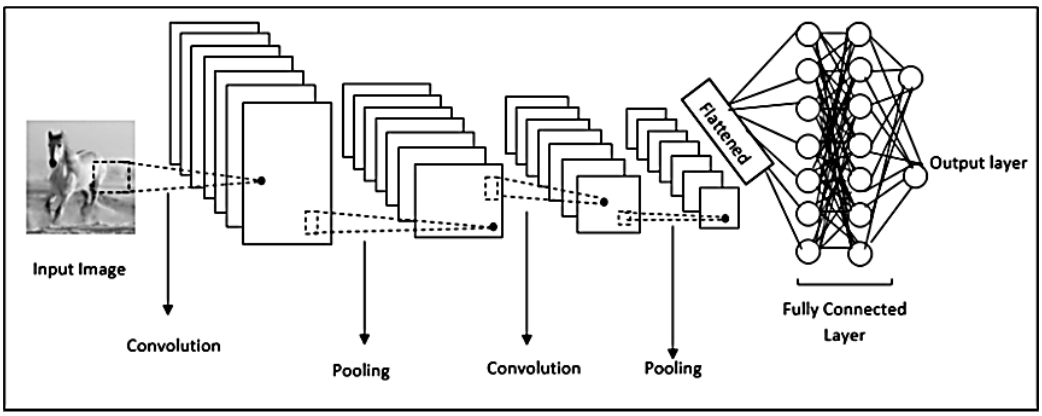


Figure1. 17:L'architecture d'un CNN.[19]

10. Les architectures de CNN :

Il existe un grand nombre de réseaux CNN pré entrainés, les plus connus sont :

Tableau 1. 1: Les architectures de CNN.

CNN	*Nombre de couches de poids*	*Top 5 classification errer*
AlexNet	8	16.4
ZFnet	8	14.8
VGG	16\18	6.8
GoogleLeNet	22	6.7
ResNet	34-152	3.5

11. Conclusion :

Dans ce chapitre on a présenté les notions importantes qui sont en relation avec l'apprentissage profond (définition, Architectures…etc.). Aussi qu'une vision générale sur l'apprentissage profond, toute on donnant en détail la méthode choisie dans notre travail de recherche qui est le CNN.

Dans le prochain chapitre, nous présentons notre approche qui traite les détails de l'attaque adverse sur la classification d'image et nous allons concentrer sur les erreurs de classification commises par les voitures autonomes

CHAPITRE II : ATTAQUES ADVERSES DE L'APPRENTISSAGE PROFOND.

1. Introduction :

Des études récentes de Google Brain ont montré que tout classificateur d'apprentissage automatique peut être trompé pour donner des prédictions incorrectes, et avec un peu de compétence, vous pouvez les amener à donner à peu près tous les résultats que vous voulez. Les réseaux de neurones profonds, sont les modèles d'apprentissage IA les plus performants Pourtant, des recherches scientifiques démontrent qu'ils peuvent être facilement dupés. On imagine alors les conséquences néfastes, voire tragiques, de ces attaques. L'omniprésence croissante a donné à des attaquants sans scrupules la possibilité d'exploiter toute vulnérabilité trouvée dans les modèles d'apprentissage automatique et les données utilisées pour les former, donnant lieu à une « IA contradictoire ». L'impact potentiel que l'IA contradictoire peut avoir sur notre société et les implications néfastes qu'elle créera pour notre sécurité, notre confiance, Ce fait devient de plus en plus déconcertant car les systèmes sont liés à l'intelligence artificielle, et nombre d'entre eux sont cruciaux pour notre vie sûre et confortable comme banques, systèmes de surveillance, distributeurs automatiques de billets, reconnaissance faciale sur votre ordinateur portable, et voitures autonomes.[20]
Dans ce chapitre, nous présentons quelques algorithmes et méthodes d'attaque adverse représentatifs. Ces méthodes visent à attaquer les modèles DL de classification d'image. Nous détaillons les attaques contradictoires spécifiques sur les autres modèles DL comme les voitures autonomes.

2. Attaque adverse :

Le terme « adversaire » est utilisé dans le domaine de la sécurité informatique pour décrire des personnes ou des machines susceptibles de tenter de pénétrer ou de corrompre un réseau ou un programme informatique. Les adversaires peuvent utiliser diverses méthodes d'attaque pour perturber un modèle d'apprentissage automatique.
L'équipe de ML à l'Université Berkeley décrit les adversaires exemples « comme étant des illusions optiques pour les machines. » Ce sont des données trompeuses, qui déjouent un l'algorithme en lui faisant croire quelque chose qui n'est pas vrai.
Dans le contexte d'algorithme de classification, un adversaire exemple représente un ensemble de données synthétiques, soigneusement élaboré pour induire une mauvaise classification.
Ces données entrées malicieusement sont la combinaison de données initialement correctement classifiées, auxquelles est ajoutée une perturbation pratiquement imperceptible.

C'est donc très difficile, pour un œil humain, de discerner si un adversaire exemple s'est glissé dans son ensemble de données.[21][20][22]

3. Terminologie associée aux adversaires exemples :

3.1. Attaques d'évasion :

La plupart des attaques pour générer des perturbations menant à des adversaires exemples se classe dans la catégorie des attaques d'évasion. Ces attaques surviennent lors de la phase d'opération de l'algorithme.

Un attaquant essaie habituellement de dissimuler du contenu malicieux pour permettre à ces échantillons d'être malgré tout catégorisés comme étant légitime. Par exemple, un attaquant pourrait vouloir déjouer un filtre antipourriel. Ainsi, en ajoutant judicieusement de faibles modifications à son pourriel, il pourrait maintenant être catégorisé comme étant un courriel légitime par le filtre.

L'attaquant n'a donc pas besoin d'avoir accès à l'algorithme durant sa phase d'entraînement.

3.2. Attaques d'empoisonnement :

Il existe également des attaques d'empoisonnement. Ces attaques sont exécutées lors de la phase d'entraînement initiale ou lors des phases de réentraînement basé sur les données recueillies en opération.

3.3. Attaque non ciblée :

Une attaque non-ciblée est simplement d'induire une mauvaise classification, peu importe la classe finale déterminée par l'algorithme. Tout ce qui importe, est que la catégorie choisie par l'algorithme ne soit pas la bonne catégorie. La bonne catégorie est celle choisie lorsque les données non-perturbées lui sont présentés.

2.4. Attaque ciblée :

Pour une attaque ciblée, l'attaquant désire que la mauvaise classification soit faite dans une catégorie en particulier.

- Par exemple :

- Une attaque pourrait permettre à un classificateur reconnaissant des nombres écrit à la main de catégoriser un chèque d'une valeur de 1 000 000$ comme étant un chèque de 9 999 999$…

  « La catégorie « chiffre neuf » serait donc la catégorie ciblée pour la catégorisation de chaque chiffre. Dans ce scénario, l'attaquant à décidément avantage à cibler une catégorie.

- Attaque : Un adversaire trompe un drone en ajoutant des déformations physiques à des objets spécifiques dans une zone.
  Résultat : le classificateur d'images du drone pense que les objets appartiennent à une catégorie spécifique prédéfinie par l'adversaire.

4. Quantité d'information nécessaires :

Les attaques sont également classées selon la quantité d'informations nécessaires à propos de l'algorithme afin de les exécutées.

4.1. Attaque boîte noire (Black Box Attack) :

L'attaquant n'a besoin d'aucune information sur l'algorithme. Il a simplement besoin d'être capable de lui fournir des données et d'observer la classification résultante.

Dans le cadre de l'attaque par la boîte noire, l'adversaire a une connaissance limitée de l'architecture du réseau neuronal profond et ne peut qu'estimer le comportement du modèle et concevoir des exemples d'adversité basés sur son estimation.

4.2. Attaque boîte grise (Gray Box Attack) :

Seulement quelques informations sont nécessaires. Exemple : les catégories possibles et les probabilités de chaque catégorie lorsque l'algorithme effectue une prédiction.

4.3. Attaque boîte blanche (White Box Attack) :

L'attaquant doit connaître l'intégralité du modèle : son architecture, son processus d'apprentissage, etc. Seul un acteur sachant tous les secrets de l'algorithme peut performer ce type d'attaque.

• d'attaque de la boîte blanche est que l'adversaire connaît parfaitement les rouages du réseau neuronal profond et peut utiliser ces connaissances pour concevoir des contributions adverses.[21]

5. Les menace graves d'intelligence artificielle :

L'intelligence artificielle est un fantastique outil quand il est au service de la santé, la technologie ou l'astrophysique. Mais dans de mauvaises mains, elle peut aussi servir à des fins criminelles ou à la désinformation.

- **Fausses vidéos :** usurper l'identité d'une personne en lui faisant dire ou faire des choses qu'elle n'a jamais dite ou faites, dans le but de demander un accès à des données sécurisées, de manipuler l'opinion onde nuire à réputation de quelqu'un...

- **Piratage de voitures autonomes :** s'emparer des commandes d'un véhicule autonome pour s'en servir comme arme (par exemple perpétrer une attaque terroriste, provoquer un accident, etc).
- **Piratage des systèmes contrôlés par l'IA :** perturber les infrastructures en causant par exemple une panne d'électricité généralisée, un engorgement du trafic ou la rupture de la logistique alimentaire.
- **Chantage à grande échelle :** recueillir des données personnelles afin d'envoyer des messages de menace automatisés. L'IA pourrait également être utilisée pour générer de fausses preuves.
- **Robots militaires :** prendre le contrôle de robots ou armes à des fins criminelles. Une menace potentiellement très dangereuse mais difficile à mettre en œuvre, le matériel militaire étant généralement très protégé.
- **Corruption de données :** modifier ou introduire délibérément de fausses données pour induire des biais spécifiques. Par exemple, rendre un détecteur insensible aux armes ou encourager un algorithme à investir dans tel ou tel marché.
- **Cyberattaque basée sur l'apprentissage :** perpétrer des attaques à la fois spécifiques et massives, par exemple en utilisant l'IA pour sonder les faiblesses des systèmes avant de lancer plusieurs attaques simultanées.
- **Drones d'attaque autonomes :** détourner des drones autonomes ou s'en servir pour s'attaquer à une cible. Ces drones pourraient être particulièrement menaçants s'ils agissent en masse dans des essaims auto-organisés.
- **Refus d'accès :** endommager ou priver des utilisateurs d'un accès à un service financier, à l'emploi, à un service public ou une activité sociale. Non rentable en soi, cette technique peut être utilisée comme chantage.
- **Reconnaissance faciale :** détourner les systèmes de reconnaissance faciale, par exemple en fabriquant de fausses photos d'identité (accès à un smartphone, caméras de surveillance, contrôle de passagers...)
- **Manipulation de marchés financiers :** corrompre des algorithmes de trading afin de nuire à des concurrents, de faire baisser ou monter une valeur artificiellement, de provoquer un crash financier….[23]

6. Attaque adverse avec FGSM (The Fast Gradient Sign Method) :

La méthode de signe de gradient rapide (FGSM) est une attaque en boîte blanche, ce qui signifie que l'attaque est générée sur la base d'une architecture réseau donnée. Le FGSM est

basé sur l'idée que les réseaux normaux suivent une descente de gradient pour trouver le point de perte le plus bas, et donc si nous suivons le signe du gradient (en allant dans la direction opposée à la descente du gradient), nous pouvons maximiser la perte en ajoutant simplement une petite quantité de perturbation.

6.1. La descente de gradient :

La descente de gradient est un algorithme d'optimisation souvent utilisé pour trouver les poids ou les coefficients des algorithmes d'apprentissage automatique, tels que les réseaux de neurones artificiels et la régression logistique.

Cela fonctionne en permettant au modèle de faire des prédictions sur les données d'apprentissage et en utilisant l'erreur sur les prédictions pour mettre à jour le modèle de manière à réduire l'erreur.

Le but de l'algorithme est de trouver des paramètres de modèle (par exemple, des coefficients ou des poids) qui minimisent l'erreur du modèle sur le jeu de données d'apprentissage. Pour ce faire, il modifie le modèle en le déplaçant le long d'une pente d'erreur vers une valeur d'erreur minimale. Cela donne à l'algorithme le nom de « descente de gradient ».

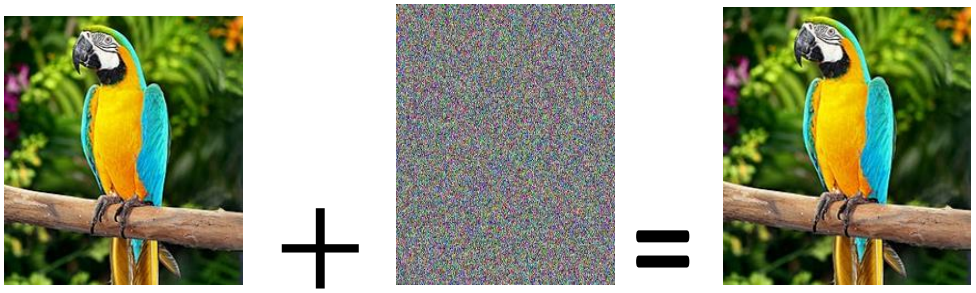
Il existe trois variantes de cette méthode :

- Batch gradient descent.
- Descente de gradient stochastique.
- Mini-batch gradient descent .[10]

Lorsque vous demandez à un humain de décrire comment il détecte un macaw dans une image, il peut rechercher des caractéristiques physiques telles que les ailes, Le bec, plumes, la queue la couleur. Elle pourrait également donner un autre contexte, comme le type d'habitat dans lequel elle s'attendrait à voir le macaw.

Pour un réseau de neurones artificiels, tant que l'exécution des valeurs de pixels à travers l'équation fournit la bonne réponse, il est convaincu que ce qu'il voit est bien un macaw. En d'autres termes, en modifiant correctement les valeurs de pixels de l'image, vous pouvez tromper l'IA en lui faisant croire qu'elle ne voit pas de macaw.

Les chercheurs en IA ont ajouté une couche de bruit à l'image. Ce bruit est à peine perceptible à l'œil humain. Mais lorsque les nouveaux numéros de pixels passent par le réseau neuronal, ils produisent le résultat attendu de l'image d'une bibliothèque.[24][25]



x 97.3% Macaw | $sign(\nabla x\, J(\theta, x, y))$ | $x + \varepsilon * sign(\nabla_x J(\theta, x, y))$ 88.9% bibliothèque

Figure2. 1: Une démonstration de la génération rapide d'exemple FGSM de confrontation appliquée sur image.

FGSM peut donc être décrit comme l'expression mathématique suivante :

$$x' = x + \varepsilon * sign(\nabla_x J(\theta, x, y))$$

$\boldsymbol{x'}$: Notre image contradictoire de sortie.

$\boldsymbol{x}$: L'image d'entrée d'origine.

$\boldsymbol{y}$: : l'étiquette de vérité terrain de l'image d'entrée.

$\boldsymbol{\varepsilon}$: Petite valeur nous multiplions les gradients signés par pour nous assurer que les perturbations sont suffisamment petites pour que l'œil humain ne puisse pas les détecter mais suffisamment grandes pour tromper le réseau neuronal.

$\boldsymbol{\theta}$: Notre modèle de réseau neuronal.

$\boldsymbol{J}$: La fonction de perte.

La création d'exemples d'apprentissage machine contradictoire est un processus d'essais et d'erreurs. De nombreux modèles d'apprentissage automatique de classificateurs d'images fournissent une liste de résultats avec leur niveau de confiance (par exemple, Macaw = 90%, perruche= 50%, pigeon= 15%, etc.).[26][27]

La création d'exemples contradictoires implique de faire de petits ajustements aux pixels de l'image et de les réexécuter via l'IA pour voir comment la modification affecte les scores de confiance. Avec suffisamment de réglages, vous pouvez créer une carte de bruit qui réduit la confiance dans une classe et l'augmente dans une autre. Ce processus peut souvent être automatisé.

N.B : La perturbation ajoutée ci-dessus peut sembler être un assortiment aléatoire de pixels, cependant, en réalité, chacun des pixels de la perturbation a une valeur (représentée par une couleur) qui est calculée à l'aide d'algorithmes mathématiques compliqués.

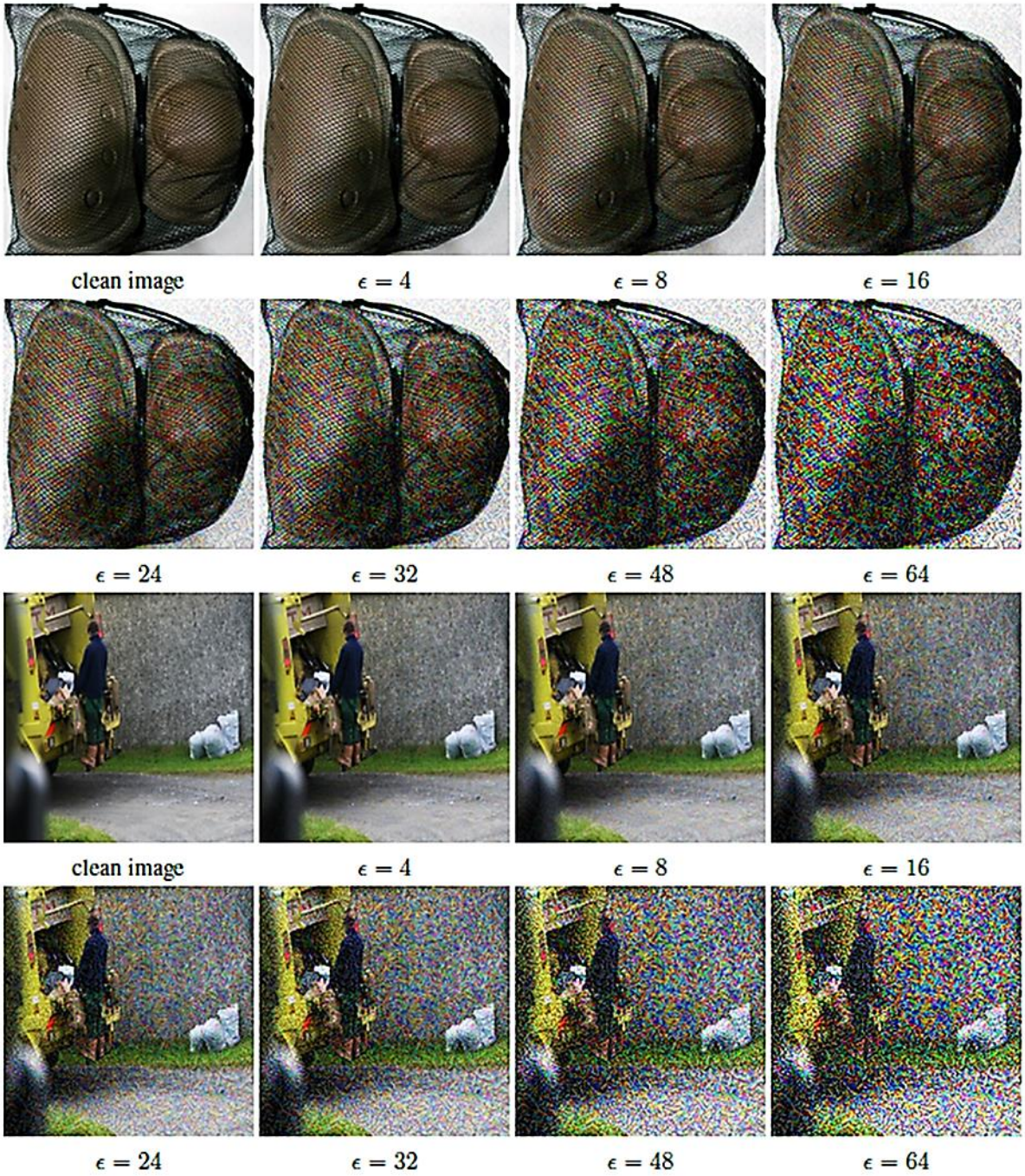


Figure 2. 2: Comparaison des images résultant d'une perturbation antagoniste selon la méthode FGSM.

6.2. Exemple algorithme de classification basé sur des séries chronologiques :

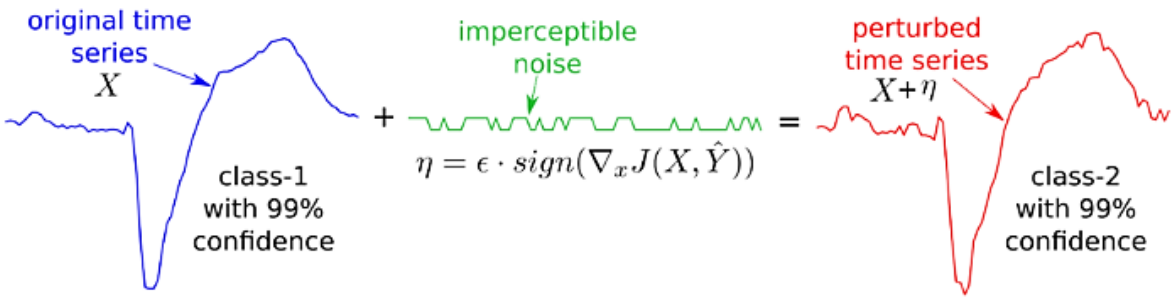


Figure2. 3: Exemple de perturbation de la classification d'une série temporelle d'entrée à partir du jeu de données TwoLeadECG en ajoutant un bruit imperceptible calculé à l'aide de la méthode des signes de gradient rapide (FGSM).

La série en bleu est correctement catégorisée dans une certaine catégorie (la classe 1) alors qu'après l'ajout d'un bruit quasi-imperceptible (en vert), l'algorithme catégorise maintenant l'exemple perturbé (en rouge) dans la classe 2.

Cet exemple pourrait représenter un classificateur qui étudie les électrocardiogrammes de patients d'un hôpital.

Un acteur malveillant pourrait donc induire une perturbation faisant passer la classification de « patient à risque de faire une crise cardiaque » à « patient en santé ».

6.3. Adversaires exemples et la voiture autonome :

un nouveau rapport de l'Agence de la cybersécurité de l'Union européenne (ENISA) suggère que les véhicules autonomes sont "très vulnérables aux attaques à grande échelle" qui pourraient présenter un risque pour les occupants. . . Et les piétons et les personnes dans d'autres véhicules. Les attaques couvertes dans le rapport incluent des attaques de capteurs utilisant des faisceaux de lumière, des systèmes de détection d'objets cassés, une activité d'arrière-plan malveillante et des attaques d'apprentissage automatique incohérentes intégrées dans les données d'exercice ou dans le monde physique.[28]

« L'attaque pourrait être utilisée pour rendre l'IA « aveugle » pour les piétons en manipulant, par exemple, le composant de reconnaissance d'image afin de mal classer les piétons. Cela pourrait entraîner des ravages dans les rues, car les voitures autonomes peuvent heurter les piétons sur la route ou les passages pour piétons », indique le rapport. « L'absence de connaissances et d'expertise suffisantes en matière de sécurité parmi les développeurs et les concepteurs de systèmes sur la cybersécurité de l'IA est un obstacle majeur qui entrave l'intégration de la sécurité dans le secteur automobile.».

Au cours des dernières années, un certain nombre d'études ont montré que les perturbations physiques peuvent tromper les systèmes de véhicules autonomes avec peu d'effort. En 2017, des chercheurs ont utilisé de la peinture en aérosol ou des autocollants sur un panneau d'arrêt pour tromper un véhicule autonome en lui faisant mal identifier le panneau comme un panneau de limitation de vitesse.

Figure 2. 4: Un simple autocollant pour tromper l'intelligence artificielle de la voiture autonome.

En 2019, les chercheurs en sécurité de Tencent ont utilisé des autocollants pour faire dévier le pilote automatique de Tesla dans la mauvaise voie. Et les chercheurs ont démontré l'année dernière qu'ils pouvaient conduire un système de véhicule autonome à accélérer rapidement de 35 mi / h à 85 mi / h en plaçant stratégiquement quelques morceaux de ruban adhésif sur la route.[21]

Figure 2.5: Autocollant pour tromper la voiture autonome pensé que la limite de vitesse était de 85 mi / h.

Simplement avec un petit autocollant ajouté au bas du panneau d'arrêt, un algorithme de reconnaissance de signalisation routière est convaincu à 97% qu'il voit un panneau de limite de vitesse.

Figure 2.6: Le panneau d'arrêt est classé à tort comme panneau de limitation de vitesse.

Malgré que les exemples contradictoires puissent sembler anodins à l'œil nu, ils peuvent être dévastateurs auprès des algorithmes de classification. Imaginez une voiture circulant à 30 km/h sur une voie express ou la réelle limite est de 120 km/h… ou l'inverse.[21][29]

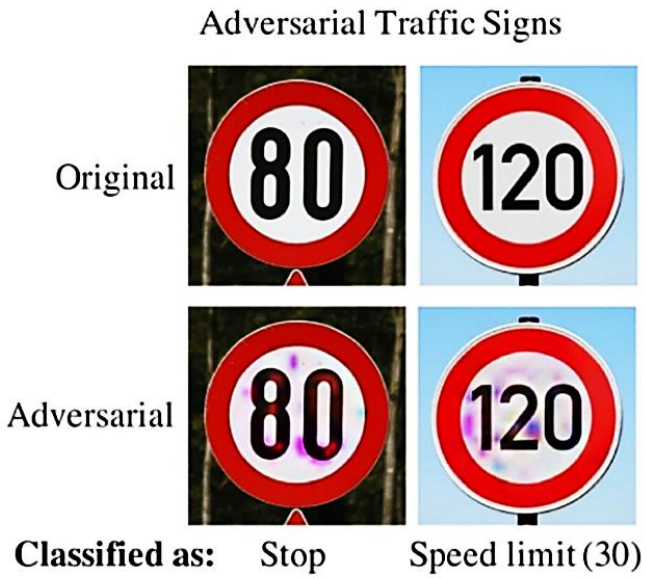


Figure2. 7: Voiture détecter 30 km/h sur autoroute ou la réelle limite est de 120 km/h.

6.4. Adversaires exemples et la reconnaissance faciale :

En 2016, des chercheurs de l'Université Carnegie Mellon ont montré que le port de lunettes imprimées spéciales en 2D pouvait tromper les réseaux de neurones de reconnaissance faciale et les confondre avec des célébrités.

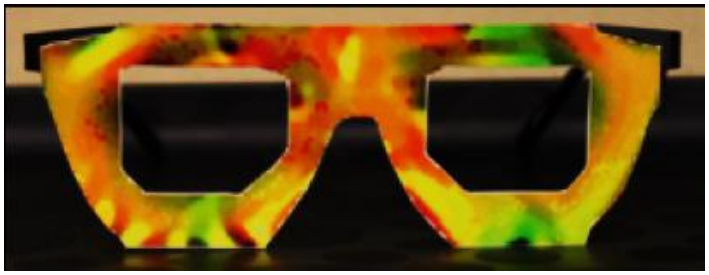


Figure2. 8: Les montures de lunettes utilisées pour éviter la reconnaissance.

Un exemple concret de personnification résultant de l'exploitation d'un adversaire exemple.

Figure2. 9: Reese Witherspoon (à gauche) portant des lunettes (au centre) imitant Russell Crowe (à droite).

À gauche, une photo de l'actrice Reese Witherspoon correctement classée par un algorithme. Au centre, la même photo, mais incluant une perturbation (les lunettes). Cette photo du centre est maintenant catégorisée par l'algorithme comme étant l'acteur Russel Crowe (à droite).[30]

En 2017 les experts en sécurité vietnamiens de Bkav, a réussi à utiliser un masque sophistiqué 3D pour tromper Face ID IPhone X et ainsi déverrouiller le téléphone, Bkav explique que le masque en question est constitué d'éléments imprimés en 3D et des images 2D qui coûte environ 200 USD.[31]

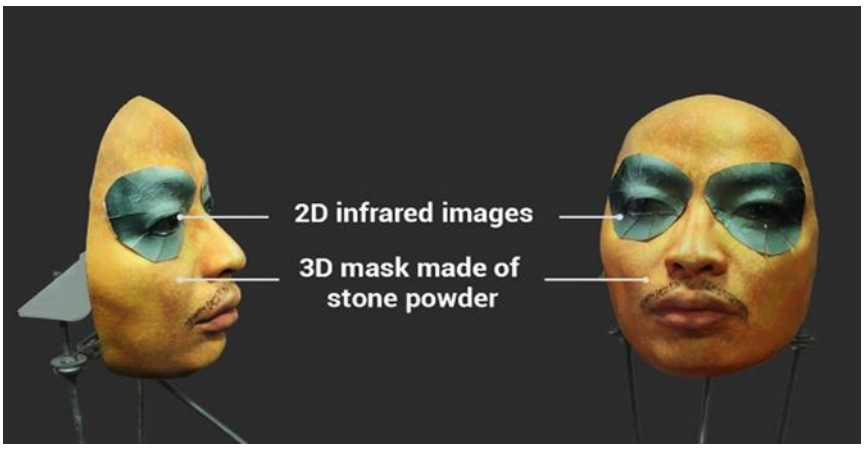


Figure2. 10: Un masque 3D élaboré pour tromper le Face ID de l'iPhone X.

7. Diagramme d'attaque :

1.On fournit des données au modèle (A) que l'on désire attaquer

2.On observe les prédictions en sortie du modèle (A).

3.On entraîne un modèle (B) pour imiter le plus fidèlement possible les prédictions modèle (A).

4.On crée des adversarial examples pour le modèle (B) en connaissant tous ses rouages internes (méthode boîte blanche).

5.On soumet le modèle (B) aux adversarial examples créés pour le modèle (A). Grâce à la propriété de transférabilité, (A) classera faussement les adversarial example qui ont été créés pour (B). Une attaque de type boîte noire vient d'être faite.

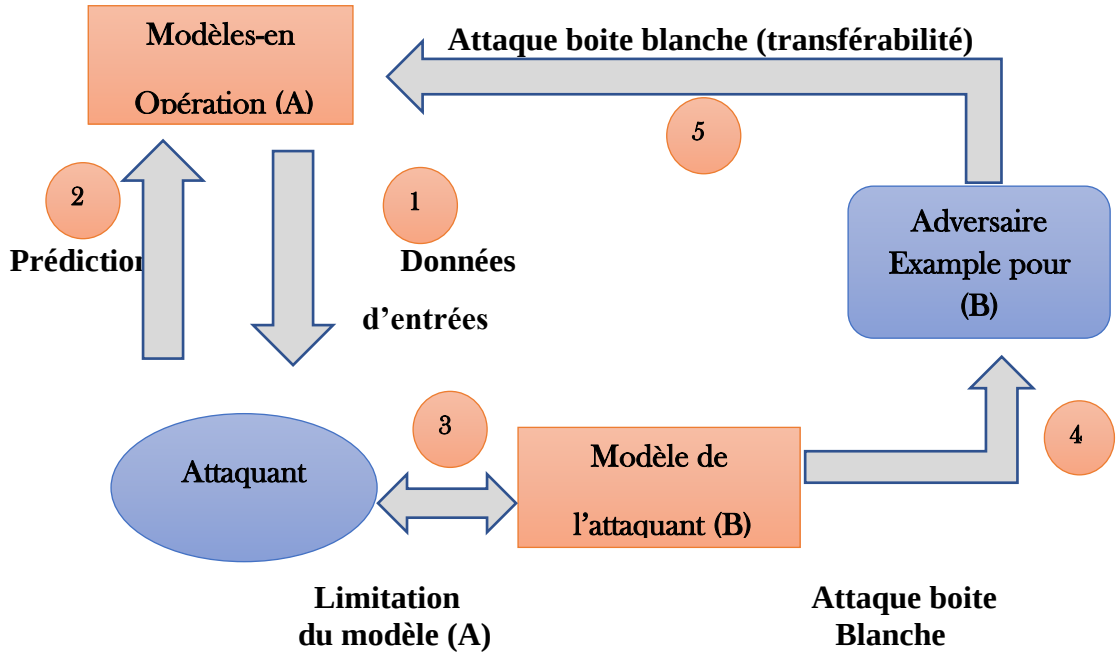


Figure2. 11: Diagramme d'attaque.[21]

8. Conclusion :

Ce chapitre a été consacré à la présentation des notions de bases du l'attaque adverse. En passant par l'explication de concepts plus général des attaques contradictoires et les impacts potentiels que l'IA contradictoire. Et aussi nous avons présenté l'état des exemples de l'IA contradictoire.

Le prochain chapitre, traite les détails de la conception, ainsi que la méthode et les outils utilisés pour la réalisation de notre application.

CHAPITRE III : CHOIX TECHNIQUE ET RÉSULTAT ÉXPÉRIMENTAUX.

1. Introduction :

Ce chapitre fait la synthèse des différents outils aussi bien logiciels que nous avons eu a utilisés tout au long de notre étude. Nous y présentons aussi les différents choix techniques que nous avons eu à faire pour mener à bien ce travail.
Enfin, on terminera avec la partie théorique, pour aborder la partie technique, dans ce chapitre contient l'implémentation du système et les résultats expérimentaux des tests.

2. Matériel :

Pour mener à bien les expérimentations, nous avons eu à utiliser des éléments aussi bien matériel que logiciel que nous détaillons dans la suite.

Hardware : Les expériences et les tests ont été effectués sur un ordinateur dont les caractéristiques sont Les suivantes :

- **Processeur :** Intel® Core™ i5-4200 M CPU @ 2.50GHz.
- **Mémoire RAM :** 4.00 Go.
- **Architecture:** 64 bits.

3. Choix techniques :

Pour notre étude, nous avons été confrontés à des choix quant au langage de programmation et aux différentes librairies à utiliser. En effet il existe plusieurs langages de programmation et bon nombre d'entre eux sont mis à jour avec l'ajout des supports sur l'apprentissage automatique.

Il est donc important de choisir judicieusement le langage à utiliser.

3.1. Python (Py) :

Python est un langage de programmation de haut niveau utilisé pour la programmation générale. Créé par Guido van Rossum et sorti en 1991, Python a une philosophie de conception qui met l'accent sur la lisibilité du code, notamment en utilisant des espaces importants. Il fournit des constructions qui permettent une programmation claire à petite et à grande échelle. Python dispose d'un système de type dynamique et d'une gestion automatique de la mémoire. Il prend en charge de multiples paradigmes de programmation, y compris orientés objet, impératifs, fonctionnels et procéduraux, et dispose d'une bibliothèque standard vaste et complète. Les interpréteurs de Py sont disponibles pour de nombreux systèmes d'exploitation. [32]

Figure 3. 1: Le logo de python.

Les outils les plus utilisés dans les sciences de données entre 2018 et 2020 a donné les résultats présents dans la figure 3.2 :

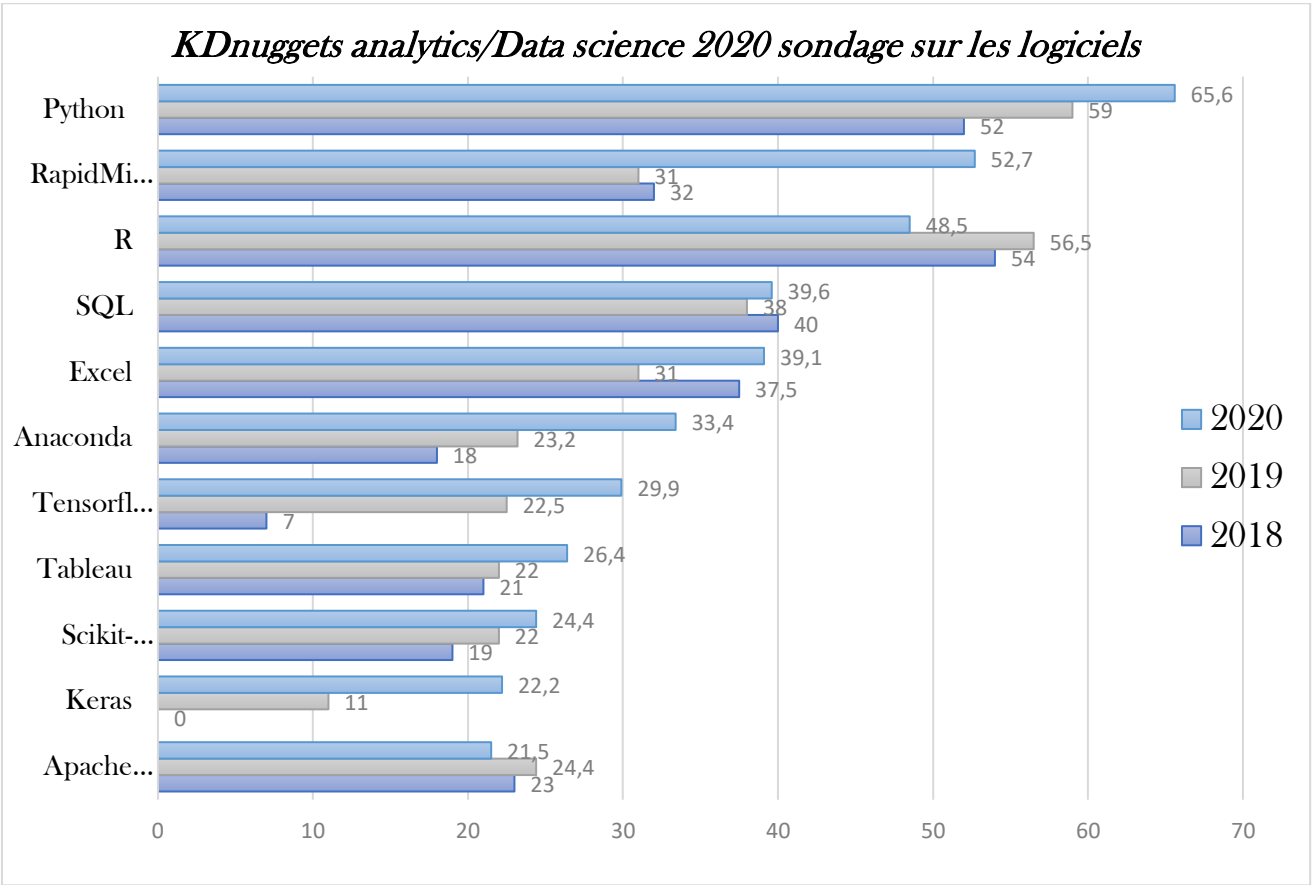


Figure 3. 2: KD nuggets Analytics / Data Science 2020. Sondage sur les logiciels : les principaux outils en 2020 et leur part dans les sondages 2018.

La figure 3.2 nous montre ainsi que depuis 2020, Python avec ses différents avantages est l'outil le plus utilisé pour le Machine Learning à la vue des différentes possibilités qu'il offre.

Notre choix après ces différentes analyses s'est donc porté sur lui en tant qu'outil principal pour mener à bien notre étude. On utilise dans notre travail python version 3.8.

En outre, il existe plusieurs distributions, paquets et modules conçus pour Py facilitant l'utilisation des concepts clés de l'apprentissage automatique. Listés sur la figure 2.1, ils se sont Avérés importants dans notre étude et sont :

• **Anaconda** : est une distribution libre et open source2 des langages de programmation Python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique (traitement de données à grande échelle, analyse prédictive, calcul scientifique), qui vise à simplifier la gestion des paquets et de déploiement3. Les versions de paquetages sont gérées par le système de gestion de paquets conda4. La distribution Anaconda est utilisée par plus de 6 millions d'utilisateurs et comprend plus de 250 paquets populaires en science des données adaptés pour Windows, Linux et MacOs.[33]

• **Spyder :** est un environnement de développement pour Python. Libre et multiplateforme (Windows, Mac OS, GNU/Linux), il intègre de nombreuses bibliothèques d'usage scientifique : Matplotlib, NumPy, SciPy et IPython.

Créé et développé par Pierre Raybaut en 2008, Spyder est maintenu, depuis 2012, par une communauté de développeurs qui ont pour point commun d'appartenir à la communauté Python scientifique.

En comparaison avec d'autres IDE pour le développement scientifique, Spyder a un ensemble unique de fonctionnalités - multiplateforme, open-source, écrit en Python et disponible sous une licence non-copyleft. Spyder est extensible avec des plugins, comprend le support d'outils interactifs pour l'inspection des données et incorpore des instruments d'assurance de la qualité et d'introspection spécifiques au code Python, tels que Pyflakes, Pylint et Rope.[8]

• **TensorFlow** : c'est une l'aide de graphiques de flux de données. Les nœuds de graphique représentent des opérations mathématiques, tandis que les arêtes de graphique représentent les tableaux de données multidimensionnels (tenseurs) qui circulent entre eux. Cette architecture flexible vous permet de déployer des calculs sur un ou plusieurs processeurs ou GPU sur un ordinateur de bureau, un serveur ou un périphérique mobile sans réécrire le code. TensorFlow inclut éga boîte à outils de visualisation de données.

TensorFlow fournit des API Python et C stables ainsi que des API rétro pour C ++, Go, Java, JavaScript et Swift.[34]

• **Keras** : est une API de réseaux de neurones de haut niveau, écrite en Python et capable de fonctionner sur TensorFlow ou Theano. Il a été développé en mettant l'accent sur l'expérimentation rapide. Être capable d'aller de l'idée à un résultat avec le moins de délai possible est la clé pour faire de bonnes recherches.

Conçu pour permettre une expérimentation rapide avec des réseaux de neurones profonds, Keras se veut simple d'utilisation, intuitive, et rapide.

Il présente un ensemble d'abstractions de niveau supérieur et plus intuitif qui facilitent la configuration des réseaux neuronaux.[35]

4. Présentation de l'application :

L'objectif principale de ce travail est de réaliser un module de l'apprentissage profond basé sur librairie de programmation d'apprentissage profond existante.
Ce module permettra la classification d'image nous choisissons modèle qui peut classer les objets on utilise les réseaux de neurones à convolution.
Nous avons utilisé des modèles pré-entraînés sur une grande quantité de données pour résoudre des problèmes difficiles. On utilise ImageNet et ResNet. Plusieurs groupes de recherche forment également de tels modèles (la formation est effectuée à l'aide de puissants GPU sur des millions d'images et d'une durée de centaines d'heures) et les rendent disponibles pour une utilisation dans des compétitions mondiales telles que le Computer Vision Large-scale Visual Recognition Challenge. Les chercheurs et les développeurs utilisent ces modèles pré-formée comme point de départ du processus de formation plutôt que de former de nouveaux modèles à partir de zéro.
Pour essayer d'obtenir un résultat rapidement, dans notre projet on imprimante une principale architecture de réseau neurone convolutif Resnet50.
Ce modèle est composé de 50 couches, et a la particularité d'introduire des connexions résiduelles. Contrairement aux réseaux de neurones convolutifs qui ont une architecture linéaire (un empilement de couches dont chaque sortie est uniquement connectée à la couche suivante [voir l'architecture A de la figure 3.3 suivante]), dans un réseau résiduel, la sortie des couches précédentes est reliée à la sortie de nouvelles couches pour les transmettre toutes les deux à la couche suivante. Un schéma s'impose [voir l'architecture B de la figure 3.3 suivante :

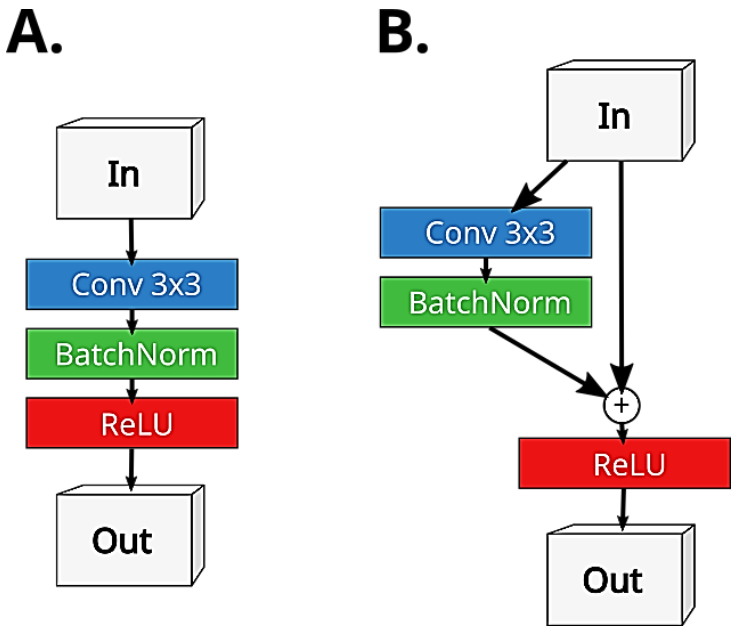


Figure 3. 3: Comparaison entre réseau neurone CNN(A) et Resnet50(B).

Avant la mise au point de ce type de réseau, il était impossible d'entraîner un réseau comportant plus de 25 couches.

Les couches devenant plus profondes, les gradients devenant plus petits, les performances en étaient inévitablement dégradées : l'erreur ne se propageait plus correctement et la mise à jour des pondérations en était directement impactée.

Les réseaux de neurones résiduels ont permis d'aller au-delà de cette limitation. Leur architecture permet la création de réseaux de neurones très profonds, à la précision meilleure que ceux ayant des architectures linéaires car ils ont la capacité d'extraire davantage d'informations et d'avoir ainsi une analyse plus avancée des images.[36]

5. Structure de programmation :

Implémentation et résultats Cette section décrit les expériences réalisées et les résultats expérimentaux de notre système sur la classification d'image.

Nous prendrons FGSM comme exemple pour illustrer le fonctionnement du ce programme Elle contient deux parties pour examiner la performance de l'information.

- **Dans la première partie**, on va commencer par créer un programme de classification d'image qui effectue une classification de base sans aucune attaque adverse.
  Ce programme démontrera que notre modèle ResNet fonctionne comme nous l'attendions (c'est-à-dire en faisant des prédictions correctes)
- **Dans cette deuxième partie**, dans ce programme, nous appliquerons des attaques aussi nous découvrirons comment construire une image contradictoire de telle sorte qu'elle perturbe ResNet.

6. Représentation du système de classification :

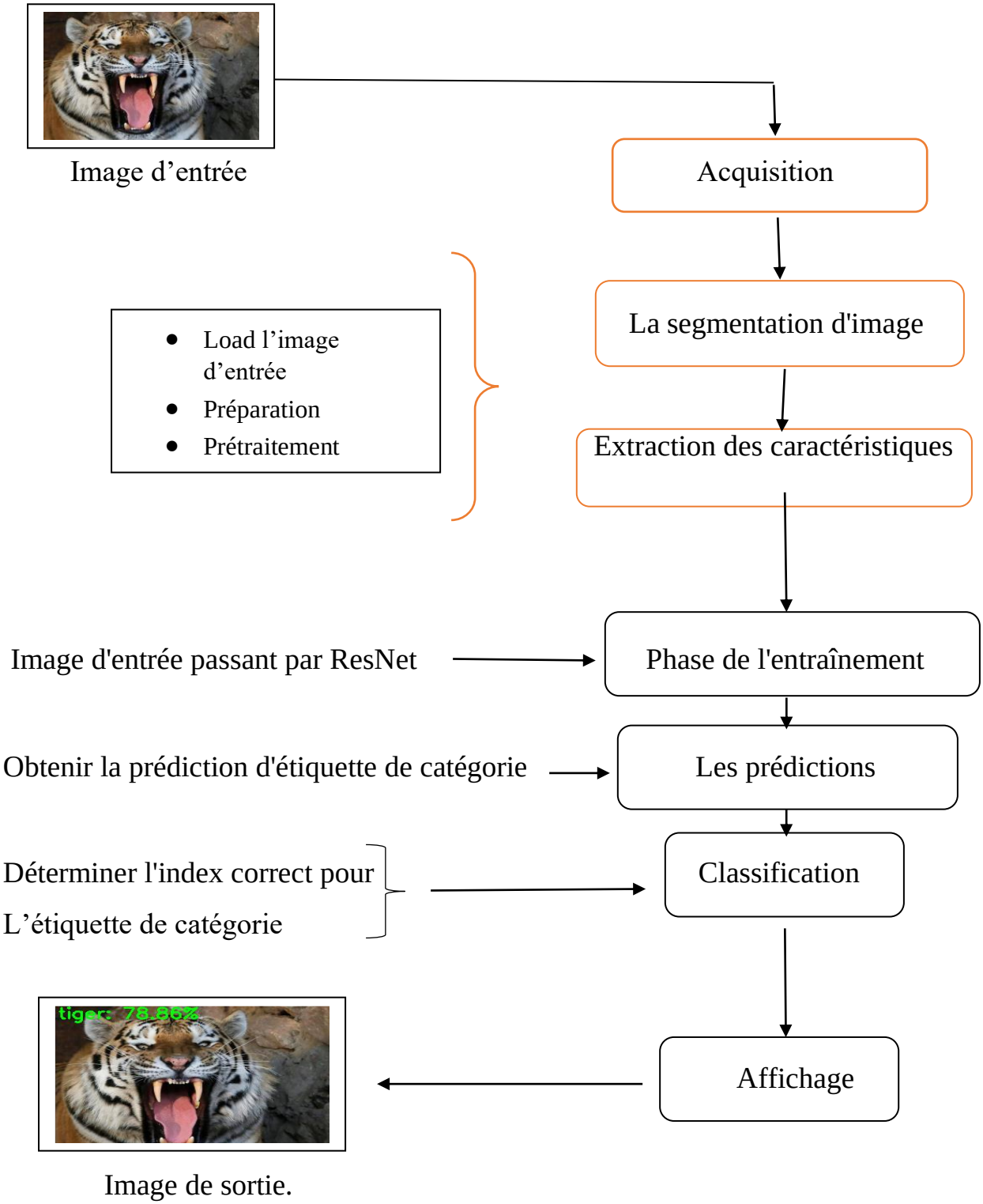


Figure 3. 4: diagramme d'un modèle du programme classification.

6.1. Acquisition :

C'est la première étape du programme où il sera entré une seule image afin de prétraiter pour être prête pour les étapes suivantes.

6.2. La segmentation d'image :

Est une opération de traitement d'image qui a pour but de rassembler des pixels entre eux suivant des critères prédéfinis. Les pixels sont ainsi regroupés en régions, qui constituent un pavage ou une partition de l'image. Il peut s'agir par exemple de séparer les objets du fond.

Prétraitement : Nous traiterons l'image en :

Basculez l'image de la disposition des canaux BGR à RVB, puis redimensionnez l'image à 224 x 224.

Il a été supposé qu'il n'y a pas de distorsion ni de biais dans l'image de texte car les images sont extraites directement de la base de données.

6.3. Extraction des caractéristiques :

L'extraction de propriétés visuelles consiste en des transformations mathématiques calculées sur les pixels d'une image numérique. Les propriétés visuelles permettent généralement de mieux interpréter certaines propriétés visuelles de l'image, que nous utiliserons pour des traitements ultérieurs dans le cadre de la détection d'objets.

6.4. Phase de l'entraînement :

La structure et les poids du modèle ResNet pré-entraîné seront chargés à partir de l'ensemble de données ImageNet.

6.5. Les prédictions :

À ce stade, à l'aide de la fonction decode_prediction, nous affichons les 3 meilleures prédictions sur notre image prétraitée et affichons le résultat de la classification sous la forme d'étiquettes de classe.

6.6. Classification :

Pour la première prédiction (c'est-à-dire prédiction a un taux de certitude plus élevé), nous afficherons l'étiquette de classification qui peut être lue sur l'écran de notre machine, puis rechercherons dans l'ensemble de Index des étiquettes de classe ImageNet à l'aide de la fonction get_class_idx.

Nous montrons également les 3 premières étiquettes et leur probabilité correspondante.

6.7. Affichage :

La dernière étape consiste à afficher la première prédiction sur l'image résultante qui s'affiche sur notre écran.

7. Résultat :

Chaque fois que nous choisissons une image d'un animal puis l'entré dans le programme afin de le classer.

1ère image:

```
[INFO] loading image...
[INFO] loading pre-trained ResNet50 model...
[INFO] making predictions...
[INFO] tiger => 292 ←——— class_idx
[INFO] 1. tiger: 78.86%
[INFO] 2. tiger_cat: 17.97%
[INFO] 3. jaguar: 2.04%
```

2ème image:

```
[INFO] loading image...
[INFO] loading pre-trained ResNet50 model...
[INFO] making predictions...
[INFO] African_crocodile => 49←——— class_idx
[INFO] 1. African_crocodile: 99.96%
[INFO] 2. American_alligator: 0.02%
[INFO] 3. alligator_lizard: 0.01%
```

3ème image:

```
[INFO] loading image...
[INFO] loading pre-trained ResNet50 model...
[INFO] making predictions...
[INFO] macaw => 88 ←——— class_idx
[INFO] 1. macaw: 99.77%
[INFO] 2. lorikeet: 0.09%
[INFO] 3. sulphur-crested_cockatoo: 0.03%
```

Comme on s'y attendait les résultats et les probabilités ont été évalués correctement. Cela est dû à l'absence des perturbations qui provoque des erreurs.

8. Représentation du système d'attaque contradictoire :

Maintenant, nous allons mener des attaques contradictoires sur le programme précédent.

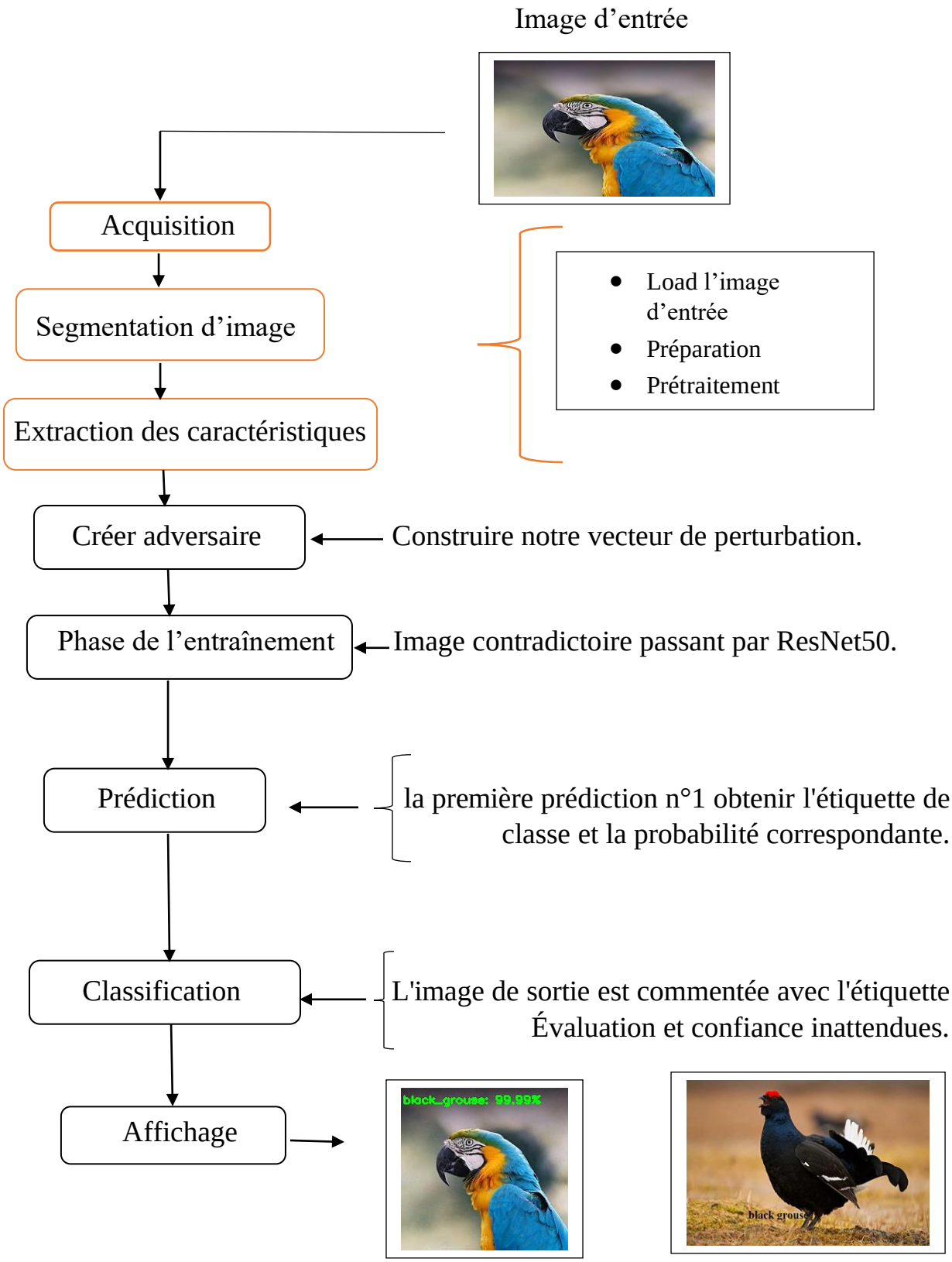


Figure 3. 5: Diagramme d'implémentation d'un modèle d'attaques contradictoires.

Nous sommes maintenant prêts à mettre en œuvre des attaques contradictoires non ciblées et à construire une image contradictoire.

Comme nous l'avons commencé dans l'exemple précédent ci-dessus, les trois premières étapes **Acquisition, Segmentation d'image et Extraction des caractéristiques** sont nécessaires pour load l'image d'entrée prétraiter redimensionner Pour être prêt après cela pour l'application des attaques adversaires.

8.1. Créer adversaire

La méthode pour crée adversaire accepte cinq paramètres obligatoires :

1.model : Notre modèle ResNet50

2. baseImage : L'image d'entrée d'origine non perturbée pour laquelle nous souhaitons construire une attaque contradictoire, ce qui fait que notre modèle la classifie de manière erronée.

3.delta : Notre vecteur de bruit, qui sera ajouté à la base Image, causant finalement l'erreur de classification. Nous mettrons à jour ce vecteur delta au moyen de la descente de gradient.

Adversaire = base Image + delta(Δ)

4.class Idx : index d'étiquette de classe d'entier de l'ensemble de données ImageNet. Nous avons obtenu cette valeur en implémentation de la première partie classant les images non contradictoires.

5.steps : nombre d'étapes de descente de gradient à effectuer (nous utilisons à 50 étapes).

8.2. Phase de l'entraînement :

Notre image contradictoire est transmise à ce ResNet50 pour mesurer et normaliser l'image conflictuelle résultante.

À partir de là, il se passe ce qui suit :

• Notre modèle fait des prédictions sur un adversaire nouvellement créé.

• La perte est calculée par rapport au class_idx d'origine (c'est-à-dire l'index complet de la première étiquette de la classe ImageNet).

• La perte résultante apparaît toutes les cinq étapes.

• Calcule les gradients de perte par rapport au vecteur de turbulence.

Enfin, nous renvoyons le vecteur de perturbation résultant à la fonction appelante, et la valeur delta finale nous permettra de construire l'attaque paradoxale utilisée pour tromper notre modèle.

8.3. Prédiction :

Dans cette étape, à l'aide de la fonction decode_prediction, nous affichons une seule prédiction c'est Top-1 prédictions sur notre image prétraitée et affichons le résultat de la classification sous la forme d'étiquettes de classe.

8.4. Classification :

L'image prétraitée résultante est transmise via ResNet, après quoi nous récupérons les 3 principales prédictions et les décodons, nous saisissons ensuite l'étiquette et la probabilité et confiance correspondante avec la prédiction top-1 et affichons ces valeurs.

8.5. Affichage :

La dernière étape consiste à dessiner la prédiction supérieure sur notre image contradictoire de sortie et à l'afficher sur notre écran.

9. Résultat:

1ère image:

```
[INFO] loading image...
[INFO] loading pre-trained ResNet50 model...
[INFO] generating perturbation...
step: 0, loss: -0.000439428084064275...
step: 5, loss: -0.0013993718894198537...
step: 10, loss: -0.026845110580325127...
step: 15, loss: -2.595684051513672...
step: 20, loss: -10.092496871948242...
step: 25, loss: -17.36520767211914...
step: 30, loss: -24.615922927856445...
step: 35, loss: -31.134613037109375...
step: 40, loss: -38.45033645629883...
step: 45, loss: -46.38182067871094...
[INFO] creating adversarial example...
[INFO] running inference on the adversarial example...
```

[INFO] label: **snow_leopard** confidence: 99.76%

2ème image:

[INFO] loading image...

[INFO] loading pre-trained ResNet50 model...

[INFO] generating perturbation...

step: 0, loss: -0.000439428084064275...

step: 5, loss: -0.0013993718894198537...

step: 10, loss: -0.026845110580325127...

step: 15, loss: -2.595684051513672...

step: 20, loss: -10.092496871948242...

step: 25, loss: -17.36520767211914...

step: 30, loss: -24.615922927856445...

step: 35, loss: -31.134613037109375...

step: 40, loss: -38.45033645629883...

step: 45, loss: -46.38182067871094...

[INFO] creating adversarial example...

[INFO] running inference on the adversarial example...

[INFO] label : **agama** confidence : 100.00%

3ème image :

```
[INFO] loading image...
[INFO] loading pre-trained ResNet50 model...
[INFO] generating perturbation...
step: 0, loss: -0.000439428084064275...
step: 5, loss: -0.0013993718894198537...
step: 10, loss: -0.026845110580325127...
step: 15, loss: -2.595684051513672...
step: 20, loss: -10.092496871948242...
step: 25, loss: -17.36520767211914...
step: 30, loss: -24.615922927856445...
step: 35, loss: -31.134613037109375...
step: 40, loss: -38.45033645629883...
step: 45, loss: -46.38182067871094...
[INFO] creating adversarial example...
[INFO] running inference on the adversarial example...
[INFO] label: black_grouse confidence: 99.99%
```

Les attaques ou tromperies sur les modèles d'intelligence artificielle sont très évolutifs en raison des modifications de l'environnement (bruits, changement de lumière...).

Néanmoins, prenons le cas d'un système de voiture autonome qui utilise les réseaux de neurones pour identifier les panneaux de signalisation routiers.
Nous prenons le cas d'un système de voiture autonome qui utilise les réseaux de neurones pour identifier les panneaux de signalisation routiers.
Mais mis dans une perspective différente, le bruit contradictoire pourrait être dangereux. Par exemple, les voitures autonomes de Tesla peuvent être amenées à penser que les autocollants sur la route comme autre chose. En tant que tel, le véhicule pourrait changer de vitesse et se diriger directement vers la circulation en approche " qui pourraient présenter un risque pour les occupants. . . Et les piétons et les personnes dans d'autres véhicules"

Par conséquent, dans cette section, nous expliquerons le degré de danger lié aux voitures autonomes**.**

```
[INFO] loading image...
[INFO] loading pre-trained ResNet50 model...
[INFO] making predictions...
INFO] street_sign => 919      ←———    class_idx
[INFO] 1. street_sign: 76.58%
[INFO] 2. traffic_light: 6.48%
[INFO] 3. chainlink_fence: 2.35%
```

```
[INFO] loading image...
[INFO] loading pre-trained ResNet50 model...
[INFO] generating perturbation...
step: 0, loss: -0.000439428084064275...
step: 5, loss: -0.0013993718894198537...
step: 10, loss: -0.026845110580325127...
step: 15, loss: -2.595684051513672...
step: 20, loss: -10.092496871948242...
step: 25, loss: -17.36520767211914...
step: 30, loss: -24.615922927856445...
step: 35, loss: -31.134613037109375...
step: 40, loss: -38.45033645629883...
step: 45, loss: -46.38182067871094...
[INFO] creating adversarial example...
[INFO] running inference on the adversarial example...
[INFO] label: spider_web confidence: 100.00%
```

Afin de clarifier davantage, l'erreur n'est pas dans un réseau neurone, mais plutôt une attaque ciblée au niveau des données d'image pour altérer la sortie et tromper le travail du programme, c'est la preuve de l'existence d'une faille dans ce type Les classifications pour chaque type de programme ont leurs propres faiblesses uniques.

10. Résultats obtenus et discussion :

La majorité des étiquettes sont prédites correctement. La partie la plus intéressante de la matrice de confusion, cependant, sont les erreurs que fait le classificateur. Il semble parfois

confondre les avions avec les oiseaux ou les automobiles avec les camions ou les chiens avec les chats. Ces trois paires de classes présentent de nombreuses similitudes. Les avions et les oiseaux ont des ailes et un corps allongé entre les deux. Les camions et les automobiles ont sans aucun doute des caractéristiques très similaires, ce qui rend difficile pour le classificateur de faire la distinction entre les deux classes. Il en va de même pour les chiens et les chats.

La formation du classificateur devrait prendre environ 7 minutes par époque. Avec les hyperparamètres sélectionnés dans le notebook, on obtient déjà de très bons résultats. Cependant, nous ne sommes même pas proches de l'état de l'art en matière de reconnaissance d'images.

Nous verrons les performances de classification en utilisant une matrice de confusion non normalisée et une matrice de confusion normalisée.

Afin de montrer les résultats obtenus pour les modèles, on illustre dans ce qui suit les résultats en termes de précision et d'erreur ainsi que les matrices de confusion.

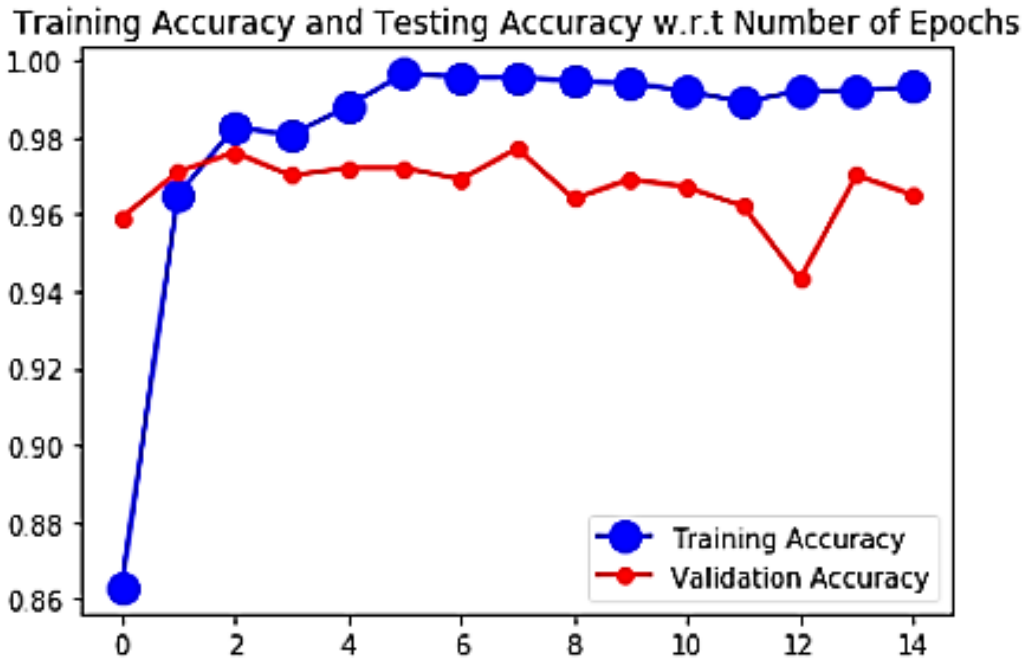


Figure 3. 6: Précision de l'entraînement et précision des tests.

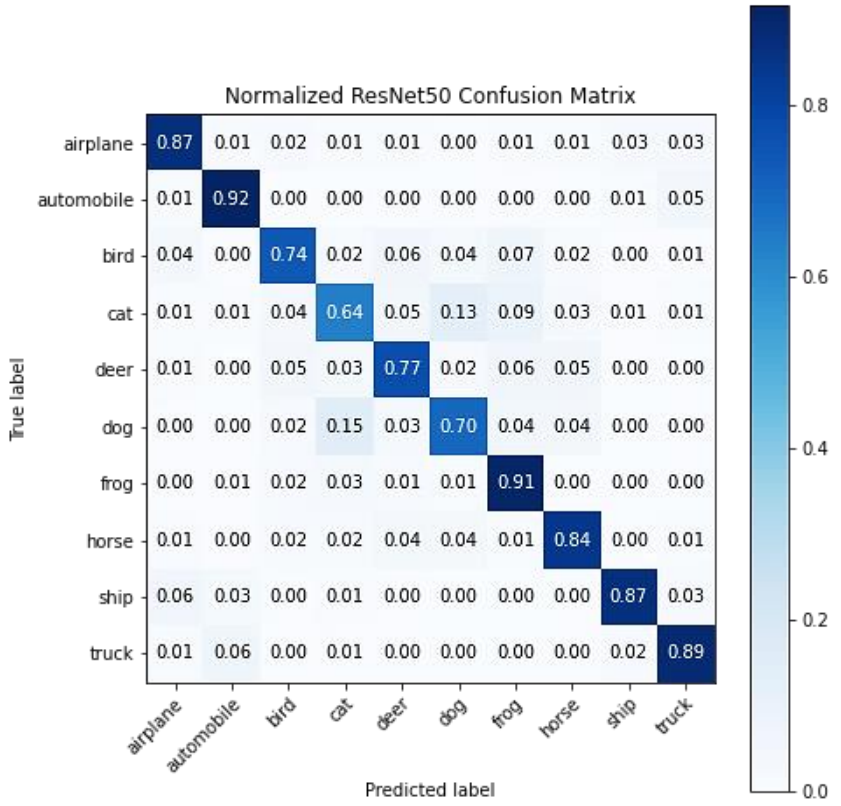


Figure 3. 7: Matrices de confusion normalisées ResNet50.

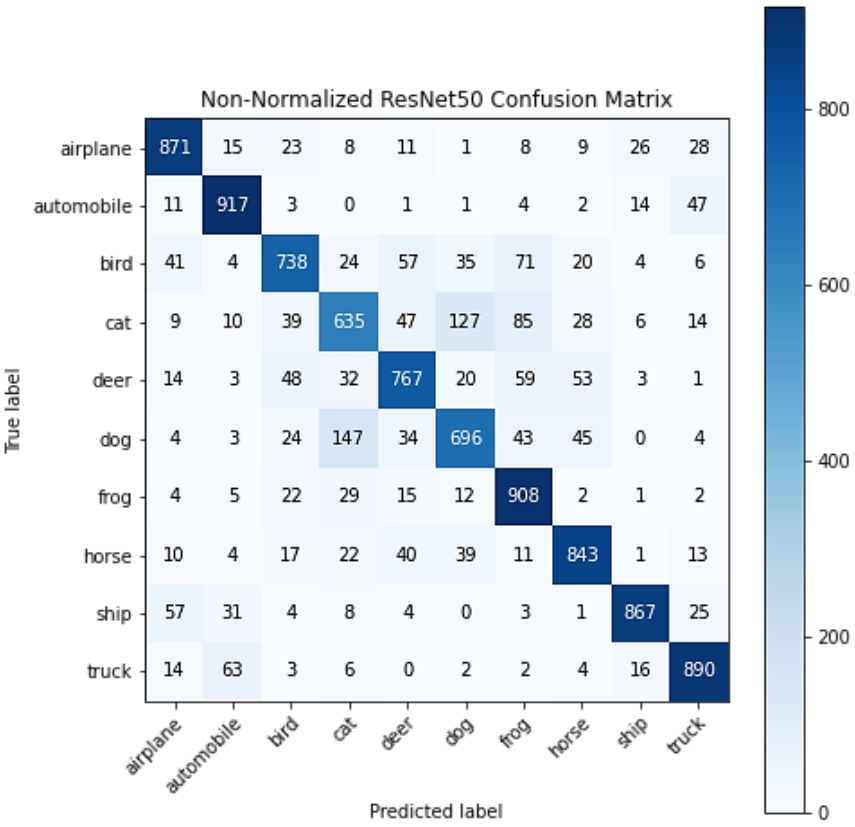


Figure 3. 8: Matrices de confusion non-normalisées ResNet50.

Comme on peut le voir ci-dessus, en analysant les matrices de confusion et le score de précision, la performance de ResNet 50 n'est pas très bonne et le score de précision moyen est de 64,8%.

On peut en déduire que la précision sera certainement améliorée si nous effectuons l'apprentissage pour un plus grand nombre d'époques. Cependant, nous avons montré l'architecture et comment implémenter les modèles. ResNet-50 a plus de paramètres à utiliser, donc évidemment, il affichera de meilleures performances

4. Les défenses adverses :

Dans le chapitre précédent, nous avons couvert des exemples contradictoires dans l'apprentissage automatique moderne, pourquoi ils sont importants et comment les générer. Nous présentons ici les méthodes de défense contradictoires utilisées pour contrer ces attaques.

4.1. Les défenses adverses :

Ce sont des techniques utilisées pour se protéger contre les attaques adverses. La course aux armements entre attaques adverses et défenses se poursuit et s'intensifie. De nombreuses méthodes de défense contradictoires ont été proposées au cours des dernières années, mais aucune d'entre elles ne garantit la sécurité contre tous les exemples et contributions contradictoires. Une défense antagoniste robuste est un domaine critique de la recherche continue, cruciale pour des solutions d'apprentissage automatique fiables.

4.2. Terminologie associée aux La défense adverse

•**Perturbation contradictoire :** la différence entre un exemple non contradictoire et son homologue contradictoire.

•**Robustesse contradictoire :** la propriété de résister à la classification erronée des exemples contradictoires.

•**Détection contradictoire :** un ensemble de méthodes pour détecter des exemples contradictoires.

•**Adversaire training :** une technique de défense dans laquelle un modèle est formé sur des exemples contradictoires. Il est important de noter que la notion d'entraînement contradictoire en tant que moyen de défense n'est pas la même que la notion d'entraînement contradictoire telle qu'utilisée dans les Générative Adversaire Networks

•**Masquage de dégradés :** la pratique de modifier un modèle pour cacher ses dégradés d'origine à un attaquant. En d'autres termes, les méthodes de défense masquent les gradients de la sortie du modèle par rapport à ses entrées.

•**Gradients obscurcis :** une forme de masquage de gradient qui englobe des gradients brisés, stochastiques, en voie de disparition et explosifs.

•**Dégradés brisés :** lorsque les gradients d'un modèle sont difficiles à calculer en raison d'opérations non différentiables.

•**Gradients stochastiques :** lorsque les gradients d'un modèle sont difficiles à calculer en raison de certaines opérations stochastiques.

•**Dégradés évanouissant :** lorsque les dégradés d'un modèle sont petits ou proches de zéro.

•**Dégradés explosifs :** lorsque les dégradés d'un modèle sont extrêmement grands ou proches de l'infini.

5. Stratégies de défense :

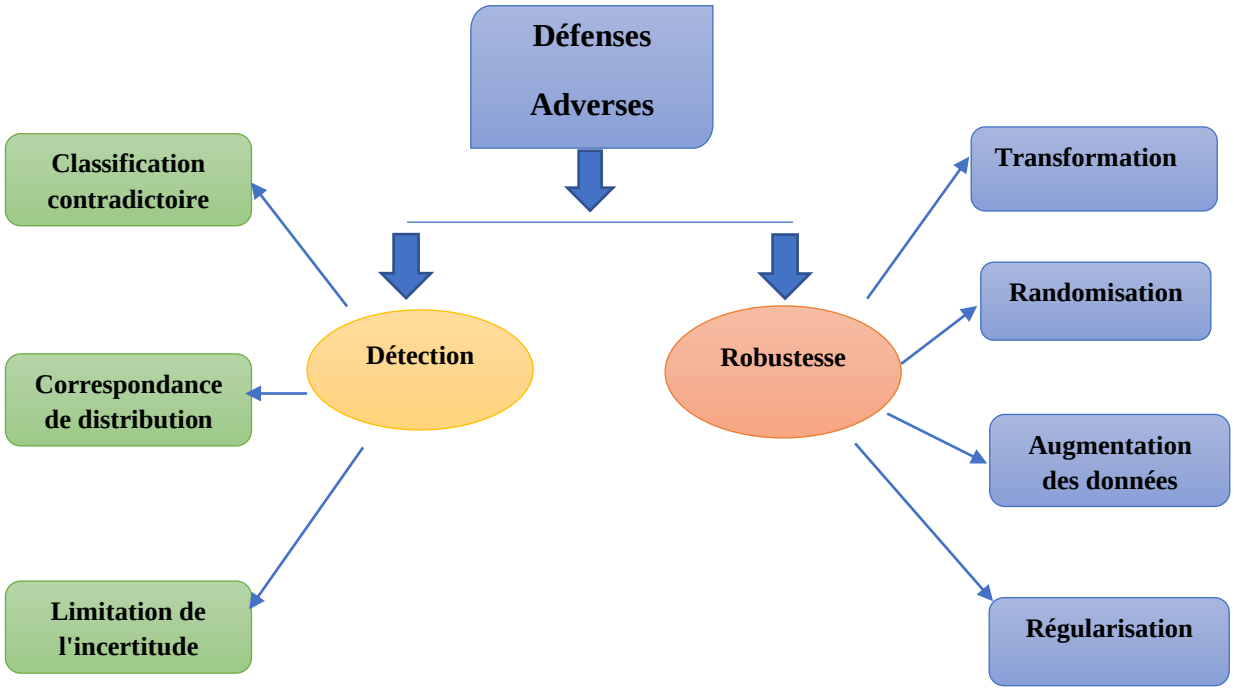


Figure 3. 9: Un graphique qui représente la stratégie de défense.

Il s'agit d'une famille de méthodes de défense qui implique une augmentation des données. L'idée de base est d'inclure des exemples contradictoires dans l'ensemble de formation. Le défenseur utilisera ces exemples pour imposer l'invariance dans la sortie du modèle étant donné l'exemple original et son homologue contradictoire. Ceci est lié aux formes classiques d'augmentation des données, dans lesquelles les données sont transformées de manière à refléter les invariances que nous souhaitons que le modèle présente.[38]

Conclusion générale :

Dans ce projet, nous avons discuté des notions fondamentales de l'apprentissage profond, des algorithmes les plus populaires, des réseaux de neurones en général et des réseaux de neurones convolutionnels en particulier. Nous avons introduit ces réseaux de neurones convolutionnels en présentant les différents types de couches utilisées dans la classification (couche convolutionnelle, couche de correction, couche de pooling et couche fullyconnected).

On a implémenté La classification automatique des images consiste à attribuer automatiquement une classe à une image à l'aide d'un système de classification.
Nous avons créé un programme qui classe les images et utilisé l'exemple de FGSM.
Au début, il classera correctement l'image avec des taux de certitude élevés comme nous l'attendions. Pour la deuxième fois, on ajoutera du bruit à une image en ajoutant des valeurs $\boldsymbol{\varepsilon}$.
Ce bruit est à peine perceptible à l'œil humain, ce qui rendra sa classification erronée.
Et nous aussi afficher à la fin la matrice de confusion pour récupérer le résultat des images bien et mal classées.
L'implémentation a été faite avec le langage de programmation python et on a utilisé des bibliothèques pour faciliter la tâche de création de nos modèles et pour l'accélération du training et enfin on a terminé avec quelques stratégies de défense.

Finalement, Nous voulions aussi préciser que La vulnérabilité des modèles de Deep Learning a été introduite et étudiée pour la première fois en 2013. Depuis, si de nombreuses plateformes open source existent pour proposer des mécanismes défensifs, cela ne suffit pas. Car parallèlement à la croissance exponentielle des modèles d'apprentissage qui surpassent de nos jours l'être humain dans de nombreux

Pour limiter la propagation et la naissance d'attaques d'un nouveau genre, il est nécessaire que les data-scientists, les laboratoires de recherche et les entreprises technologiques prennent en main et communiquent sur ce sujet autant que sur l'éthique ou l'explicabilité. Les acteurs de l'IA se doivent d'échanger et de partager les bonnes pratiques qui favoriseront la construction de modèles robustes et de mécanismes de défense qui ont fait leur preuve.Car sans mécanisme de défense, même la plus évoluée des intelligences artificielles n'a aucun avenir.

Bibliographie :

[1] M. Zouinar, "Évolutions de l'Intelligence Artificielle : quels enjeux pour l'activité humaine et la relation Humain‑Machine au travail ?," *Activites*, no. 17–1, Apr. 2020, doi: 10.4000/activites.4941.

[2] "La révolution de l'intelligence artificielle | ICI Radio-Canada.ca." https://ici.radio-canada.ca/nouvelles/special/2017/02/intelligence-artificielle/voir-vision-apprentissage-profond-reseau-neurone.html (accessed May 30, 2021).

[3] M. Smith, "Intelligence artificielle et développement humain."

[4] "Comment le « deep learning » révolutionne l'intelligence artificielle." https://www.lemonde.fr/pixels/article/2015/07/24/comment-le-deep-learning-revolutionne-l-intelligence-artificielle_4695929_4408996.html (accessed May 28, 2021).

[5] U. Kasdi and M. Ouargla, "UNIVERSITE KASDI MERBAH OUARGLA Faculté des Nouvelles Technologies de l'Information et de la Communication Département d'Informatique et des Technologies de l'information L`apprentissage profond (Deep Learning) pour la classification et la recherche d'ima."

[6] D. Bouadi, "Thème : Classification d ' images agricoles avec le Deep Learning Realisé par : Encadré par : Mme Rachida AOUDJIT Tables des matières : Chapitre 2 : Deep Learning , Machine Learning et Reseaux de neurones et," 2019.

[7] "Comprendre le DEEP LEARNING - une introduction aux réseaux de neurones de Jean-Claude Heudin-201710121430.pdf." .

[8] A. Habba and O. Ishak, "La classification des images satellitaires par l ' apprentissage profonde (deep learning)," 2019.

[9] "Apprentissage automatique et Apprentissage profond | STEMMER IMAGING." https://www.stemmer-imaging.com/fr-ch/conseil-techniqu/apprentissage-automatique-et-apprentissage-profond/ (accessed May 28, 2021).

[10] D. Nene, A. Dian, and K. M. Nadjib, "Mémoire de Fin d ' études Master La reconnaissance des expressions faciales," 2019.

[11] H. Fethallah, M. Mohammed, B. Akkacha, and S. M. Ismail, "Classification des images par les réseaux de neurones," 2017, [Online]. Available: http://dspace.univ-tlemcen.dz/bitstream/112/12235/1/Classification-des-images-avec-les-reseaux-de-neurones.pdf.

[12] "Machine learning, deep learning : quelles différences ?" https://siecledigital.fr/2019/01/30/differences-intelligence-artificielle-machine-learning-deep-learning/ (accessed May 28, 2021).

[13] "A History of Machine Learning and Deep Learning | Import.io." https://www.import.io/post/history-of-deep-learning/ (accessed May 30, 2021).

[14] "Deep Learning ou apprentissage profond : définition, concept." https://www.lebigdata.fr/deep-learning-definition (accessed May 30, 2021).

[15] Q. B. Baloch, “No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title,” vol. 11, no. 1, pp. 92–105, 2017.

[16] “MINISTÈRE DE L’ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE Université Mouloud Mammeri de Tizi-ouzou.”

[17] Y. D. Moualek, “Deep Learning pour la classification des images,” pp. 2016–2017, 2017.

[18] “Qu’est ce qu’un réseau de neurones convolutif (ou CNN) ? - Classez et segmentez des données visuelles - OpenClassrooms.” https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5082166-quest-ce-quun-reseau-de-neurones-convolutif-ou-cnn (accessed May 28, 2021).

[19] S. Ravichandiran, *Hands-On Deep Learning*. 2019.

[20] “How Adversarial Attacks Work.” https://blog.ycombinator.com/how-adversarial-attacks-work/ (accessed May 28, 2021).

[21] “Adversarial examples : validez vos modèles de Deep Learning | Moov AI.” https://moov.ai/fr/blog/validation-modeles-adversarial-examples/ (accessed May 30, 2021).

[22] “Attacking Machine Learning with Adversarial Examples.” https://openai.com/blog/adversarial-example-research/ (accessed May 30, 2021).

[23] “Les 20 menaces les plus dangereuses de l’intelligence artificielle.” https://www.futura-sciences.com/tech/questions-reponses/intelligence-artificielle-20-menaces-plus-dangereuses-intelligence-artificielle-14343/ (accessed May 30, 2021).

[24] “Adversarial Attack and Defense on Neural Networks in PyTorch | by Ta-Ying Cheng | Towards Data Science.” https://towardsdatascience.com/adversarial-attack-and-defense-on-neural-networks-in-pytorch-82b5bcd9171 (accessed May 30, 2021).

[25] “What is adversarial machine learning? – TechTalks.” https://bdtechtalks.com/2020/07/15/machine-learning-adversarial-examples/ (accessed May 30, 2021).

[26] X. Yuan, P. He, Q. Zhu, and X. Li, “Adversarial Examples: Attacks and Defenses for Deep Learning,” *IEEE Trans. neural networks Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, 2019, doi: 10.1109/TNNLS.2018.2886017.

[27] K. Ren, T. Zheng, Z. Qin, and X. Liu, “Adversarial Attacks and Defenses in Deep Learning,” *Engineering*, vol. 6, no. 3, pp. 346–360, Mar. 2020, doi: 10.1016/j.eng.2019.12.012.

[28] “EU report warns that AI makes autonomous vehicles vulnerable to attacks.” .

[29] H. Xu *et al.*, “Adversarial Attacks and Defenses in Images, Graphs and Text: A Review,” doi: 10.1007/s11633-019-1211-x.

[30] S. Chan, “ECE595 / STAT598 : Machine Learning I Lecture 33 Adversarial Attack : An Overview Today ’ s Agenda,” 2020, [Online]. Available: https://engineering.purdue.edu/ChanGroup/ECE595/video.html.

[31] “iPhone X : Face ID d’Apple trompé par un masque imprimé en 3D - ZDNet.” https://www.zdnet.fr/actualites/iphone-x-face-id-d-apple-trompe-par-un-masque-

imprime-en-3d-39860760.htm (accessed May 30, 2021).

[32] “Welcome to Python.org.” https://www.python.org/ (accessed May 30, 2021).

[33] “Anaconda | Individual Edition.” https://www.anaconda.com/products/individual (accessed May 30, 2021).

[34] “TensorFlow.” https://www.tensorflow.org/ (accessed May 30, 2021).

[35] “Keras: the Python deep learning API.” https://keras.io/ (accessed May 30, 2021).

[36] “Détecter des formes dans des photos de paysage — Makina Corpus.” https://makina-corpus.com/blog/metier/2019/detecter-des-formes-dans-des-photos-satellites (accessed May 30, 2021).

[37] P. Fuchs and P. Poulain, “Cours de Python,” p. 87, 2011.

[38] “Sécurisation des modèles d’apprentissage automatique contre les attaques adverses.” .